



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI LANSIA CARE UNTUK MONITORING LANSIA YANG TINGGAL SENDIRI BERBASIS ANDROID

**ANNISA ARUM MUMTAZAH
NRP 5111 100 089**

**Dosen Pembimbing I
Dwi Sunaryono, S.Kom., M.Kom.**

**Dosen Pembimbing II
Abdul Munif, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT - KI141502

DESIGN AND IMPLEMENTATION OF LANSIA CARE ANDROID APPLICATION FOR MONITORING THE ELDERLY WHO LIVE ALONE

**ANNISA ARUM MUMTAZAH
NRP 5111 100 089**

**Supervisor I
Dwi Sunaryono, S.Kom., M.Kom.**

**Supervisor II
Abdul Munif, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015**

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI LANSIA CARE UNTUK MONITORING LANSIA YANG TINGGAL SENDIRI BERBASIS ANDROID

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Algoritma dan Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

ANNISA ARUM MUMTAZAH

NRP. 5111 100 089

Disetujui oleh Dosen Pembimbing

1. Dwi Sunaryono, S.Kom., M.Kom.
NIP: 197205281997021001
(Pembimbing 1)
2. Abdul Munif, S.Kom., M.Sc.
NIP: 5100201301005
(Pembimbing 2)



**SURABAYA
JUNI, 2015**

RANCANG BANGUN APLIKASI LANSIA CARE UNTUK MONITORING LANSIA YANG TINGGAL SENDIRI BERBASIS ANDROID

Nama : Annisa Arum Mumtazah
NRP : 5111100089
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Dwi Sunaryono, S.Kom., M.Kom.
Dosen Pembimbing II : Abdul Munif, S.Kom., M.Sc.

Abstrak

Seiring meningkatnya derajat kesehatan dan kesejahteraan penduduk akan berpengaruh pada peningkatan Usia Harapan Hidup (UHH) di Indonesia yang menyebabkan jumlah penduduk lanjut usia (lansia) semakin meningkat dari tahun ke tahun. Secara global, kebanyakan lansia hidup mandiri, apakah dengan status hidup sendiri atau dengan pasangannya. Hanya $\pm 5\%$ yang hidup di panti. Sekitar 65% lansia yang mengalami gangguan kesehatan hidup ditemani oleh seseorang yang mengingat masalah kesehatannya. Lansia yang tinggal sendiri rentan mengalami masalah baik fisik maupun psikologis. Masalah-masalah tersebut diantaranya adalah rentannya lansia untuk terjatuh, ketakutan akan kematian, pengelihan yang berkurang, kepikunan, dan rasa terisolasi. Masalah tersebut dapat berkembang menjadi keadaan darurat apabila tidak segera ditangani.

Untuk mengatasi masalah tersebut, dibangun sebuah aplikasi Android yang menjembatani hubungan antara pengasuh dengan lansia. Aplikasi Android ini dapat mendeteksi apakah lansia terjatuh dengan mengklasifikasi data sensor

accelerometer dan gyroscope pada perangkat keras Android dan mengirimkan pesan darurat kepada pengasuh melalui SMS. Selain itu, aplikasi ini juga memfasilitasi lansia untuk melaporkan status darurat dengan menggunakan perintah suara (Speech recognition) kepada pengasuh, dan menggunakan text-to-speech untuk mengingatkan lansia waktu pemeriksaan kesehatan dan waktu minum obat serta untuk menanyakan keadaan lansia pada waktu yang telah ditentukan dan mengirimkan laporannya kepada pengasuh melalui SMS.

Kata kunci: *Android, klasifikasi k-Nearest Neighbour, lansia, sensor accelerometer dan gyroscope, speech recognition, text-to-speech*

DESIGN AND IMPLEMENTATION OF LANSIA CARE ANDROID APPLICATION FOR MONITORING THE ELDERLY WHO LIVE ALONE

Name	: Annisa Arum Mumtazah
NRP	: 5111100089
Department	: Informatics Engineering, FTIf, ITS
Supervisor I	: Dwi Sunaryono, S.Kom., M.Kom.
Supervisor II	: Abdul Munif, S.Kom., M.Sc.

Abstract

As the level of health and well-being of the population increase, it will affect on the increase of life expectancy in Indonesia, which caused the number of elderly population is increasing year by year. Globally, most of the elderly live independently, whether living alone or with their partner. Only $\pm 5\%$ elderly live in nursing home. Approximately 65% of the elderly who experience health problems live accompanied by someone who cares of their health problems. The elderly who live alone susceptible to both physical and psychological problems. These problems include the tendency of fall, fear of death, reduced vision, dementia, and a sense of isolation. Those problems can develop into emergencies if it is not promptly treated.

To overcome those problems, an Android application built to bridge the relationship between elderly who live alone with their caregiver. This Android application can detect whether the elderly fell by classifying accelerometer and gyroscope sensor data on Android hardware and send an emergency message to the caregiver via SMS. In addition, this application also facilitate the elderly to make the emergency call by using voice commands to the caregiver, and use text-to-speech to remind elderly the time of

health check up and time to take the medicine and also ask the elderly about their health condition in given time and report it to the caregiver via SMS.

Keywords: accelerometer and gyroscope sensor, Android, elderly, k-Nearest Neighbour classification, speech recognition, text-to-speech

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis, sehingga tugas akhir berjudul “Rancang Bangun Aplikasi Lansia *Care* untuk *Monitoring* Lansia yang Tinggal Sendiri Berbasis Android” ini dapat selesai sesuai dengan waktu yang telah ditentukan.

Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku tugas akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT, karena berkat ridhaNya lah penulis dapat menyelesaikan tugas akhir ini.
2. Ayah, Mama, dan Kakak penulis yang telah memberikan dukungan moral dan material serta doa, semangat, dan motivasi yang tak terhingga untuk penulis.
3. Bapak Dwi Sunaryono selaku dosen pembimbing 1 yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
4. Bapak Abdul Munif selaku dosen pembimbing 2 yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
5. Ibu Handayani Tjandrasa selaku dosen wali penulis yang telah memberikan nasehat dan bimbingan selama masa perkuliahan.
6. Segenap dosen Teknik Informatika ITS yang telah memberikan ilmunya.
7. Teman-teman angkatan 2011 yang telah memberi penulis semangat selama masa perkuliahan di Teknik Informatika ITS.

8. Sahabat penulis sejak berjuang di jurusan Teknik Informatika ITS, Rury, Septy, Fitri, Tini, dan Pundi, yang selalu memberikan dukungan dan keceriaan selama menjalankan perkuliahan.
9. Teman seperjuangan RMK Algoritma Pemrograman, Monika, Helmy, Baskara, Risma, Ghani, Novandi, Tegar, Fahmi, Ajong yang telah memotivasi penulis dalam pengerjaan tugas akhir ini.
10. Serta semua pihak yang turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2015

DAFTAR ISI

LEMBAR PENGESAHAN	v
Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxv
1. BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
2. BAB II TINJAUAN PUSTAKA	7
2.1 Sensor.....	7
2.1.1. <i>Sensor Accelerometer</i>	8
2.1.2. <i>Sensor Gyroscope</i>	8
2.2 <i>Activity Recognition</i>	9
2.2.1. <i>Data Collection</i>	9
2.2.2. <i>Feature Extraction</i>	9
2.2.3. <i>Data Interpretation</i>	9
2.3 Sistem Operasi Android	10
2.3.1. Komponen Aplikasi Android	10

2.3.2.	<i>Activity</i>	10
2.3.3.	<i>Service</i>	11
2.3.4.	<i>Content Provider</i>	11
2.3.5.	<i>Broadcast Receiver</i>	11
2.3.6.	<i>SensorEventListener</i>	12
2.3.7.	<i>RecognitionListener</i>	12
2.3.8.	<i>LocationListener</i>	13
2.3.9.	<i>SmsManager</i>	13
2.4	<i>Text to Speech</i>	14
2.5	<i>Speech Recognition</i>	15
2.6	<i>Short Message Service (SMS)</i>	15
2.7	<i>Global Positioning System (GPS)</i>	15
2.8	<i>Training Set</i>	16
2.8.1.	<i>Data Training</i>	16
2.8.2.	<i>Rata-Rata</i>	17
2.8.3.	<i>Simpangan Baku</i>	17
2.9	<i>Java Machine Learning (Java-ML)</i>	18
2.10	<i>Klasifikasi</i>	18
2.11	<i>k-Nearest Neighbour</i>	18
3.	BAB III PERANCANGAN PERANGKAT LUNAK	25
3.1	<i>Deskripsi Umum</i>	25
3.2	<i>Arsitektur Sistem</i>	25
3.3	<i>Karakteristik Pengguna</i>	27
3.4	<i>Spesifikasi Kebutuhan Perangkat Lunak</i>	29
3.4.1.	<i>Tahapan Spesifikasi Kebutuhan</i>	29
3.4.2.	<i>Kebutuhan Fungsional</i>	30

3.4.3. Kebutuhan Non Fungsional	31
3.3 Diagram Kasus Penggunaan.....	32
3.4 Diagram Alir Sistem.....	33
3.4.1. Diagram Alir Teknik <i>Windows Sampling</i> dan <i>Overlapping</i>	34
3.4.1. Diagram Alir Teknik Deteksi Jatuh.....	36
3.4.2. Diagram Alir Proses Panggilan Darurat	38
3.5 Diagram Aktivitas Sistem	40
3.5.1. Diagram Aktivitas Melakukan Panggilan Darurat	40
3.5.2. Diagram Aktivitas Melakukan Konfirmasi Deteksi Jatuh	41
3.5.3. Diagram Aktivitas Melihat Peningat Waktu Periksa Kesehatan	41
3.5.4. Diagram Aktivitas Melihat Peningat Waktu Minum Obat	42
3.5.5. Diagram Aktivitas Melakukan Konfirmasi Keadaan Saat Ini	43
3.5.6. Diagram Aktivitas Menerima Laporan Lansia Melalui SMS.....	43
3.5.7. Diagram Aktivitas Mengatur Waktu Periksa Kesehatan Lansia	44
3.5.8. Diagram Aktivitas Mengatur Waktu Minum Obat Lansia	45
3.5.9. Diagram Aktivitas Mengatur Durasi Konfirmasi Keadaan	46
3.6 Rancangan Antarmuka	46
3.6.1. Rancangan Antarmuka Halaman <i>Log In</i>	47
3.6.2. Rancangan Antarmuka Halaman <i>Sign Up</i>	47

3.6.3.	Rancangan Antarmuka Halaman Utama	48
3.6.4.	Rancangan Antarmuka Halaman Pengaturan	49
3.6.5.	Rancangan Antarmuka Halaman Atur Obat	49
3.6.6.	Rancangan Antarmuka Halaman Atur Jadwal	50
3.6.7.	Rancangan Antarmuka Halaman Ubah Profil	51
3.6.8.	Rancangan Antarmuka Halaman Tambah Waktu Minum Obat	51
3.6.9.	Rancangan Antarmuka Halaman Tambah Jadwal Pemeriksaan Kesehatan Lansia	52
3.6.10.	Rancangan Antarmuka Notifikasi Waktu Minum Obat	53
3.6.11.	Rancangan Antarmuka Notifikasi Jadwal Pemeriksaan Kesehatan	53
3.6.12.	Rancangan Antarmuka Notifikasi Konfirmasi Keadaan	54
3.6.13.	Rancangan Antarmuka Notifikasi Konfirmasi Jatuh	55
4.	BAB IV IMPLEMENTASI	57
4.1	Lingkungan Implementasi	57
4.1.1.	Lingkungan Perangkat Keras	57
4.1.2.	Lingkungan Perangkat Lunak	58
4.2	Implementasi Proses	58
4.2.1.	Implementasi Inisialisasi <i>SensorListener</i>	58
4.2.2.	Implementasi Mendapatkan Data Sensor <i>Accelerometer</i> dan <i>Gyroscope</i>	59
4.2.3.	Implementasi Teknik <i>Windows Sampling</i> dan <i>Overlapping</i>	60
4.2.4.	Implementasi Menghitung Rataan Data Sensor	60
4.2.5.	Implementasi Menghitung Simpangan Baku Data Sensor	61

4.2.6.	Implementasi Membuat <i>Classifier</i>	62
4.2.7.	Implementasi Klasifikasi Data Sensor.....	62
4.2.8.	Implementasi Mendapatkan Lokasi Pengguna	63
4.2.9.	Implementasi Mendaftarkan Peningat Minum Obat	64
4.2.10.	Implementasi Mendaftarkan Peningat Pemeriksaan Kesehatan	65
4.2.11.	Implementasi Mengatur Durasi Konfirmasi Keadaan	67
4.2.12.	Implementasi <i>Text-to-Speech</i> untuk Notifikasi	68
4.2.13.	Implementasi Melakukan Panggilan Darurat dengan <i>SpeechRecognition</i>	69
4.2.14.	Implementasi Pengiriman Pesan Singkat Kepada Pengasuh	70
4.3	Implementasi Antarmuka	71
4.3.1.	Implementasi Antarmuka Menu Awal	72
4.3.2.	Implementasi Antarmuka Menu Login	73
4.3.3.	Implementasi Antarmuka Menu <i>Sign Up</i>	74
4.3.4.	Implementasi Antarmuka Menu Utama	75
4.3.5.	Implementasi Antarmuka Menu Pengaturan	75
4.3.6.	Implementasi Antarmuka Menu Atur Obat	77
4.3.7.	Implementasi Antarmuka Menu Atur Jadwal.....	79
4.3.8.	Implementasi Antarmuka Menu Ubah Profil	80
4.3.9.	Implementasi Antarmuka Menu Tambah Waktu Minum Obat	81
4.3.10.	Implementasi Antarmuka Menu Tambah Waktu Pemeriksaan Kesehatan	82
4.3.11.	Implementasi Antarmuka Notifikasi Waktu Minum Obat	82

4.3.12. Implemetasi Antarmuka Notifikasi Waktu Pemeriksaan Kesehatan	83
4.3.13. Implementasi Antarmuka Notifikasi Konfirmasi Keadaan	84
4.3.14. Implementasi Antarmuka Notifikasi Konfirmasi Terjatuh	85
5. BAB V UJI COBA DAN EVALUASI	87
5.1 Lingkungan Uji Coba Aplikasi	87
5.2 Skenario Uji Coba	87
5.2.1. Uji Coba Fungsionalitas	88
5.2.2. Uji Coba Kegunaan	111
5.2.3. Uji Coba Akurasi Deteksi Jatuh	119
5.3 Evaluasi	123
5.3.1. Evaluasi Pengujian Fungsionalitas	123
5.3.2. Evaluasi Pengujian Kegunaan	124
5.3.3. Evaluasi Pengujian Akurasi	125
6. BAB VI KESIMPULAN DAN SARAN	127
6.1. Kesimpulan	127
6.2. Saran	128
7. DAFTAR PUSTAKA	129
LAMPIRAN A	133
LAMPIRAN B	137
BIODATA PENULIS	153

DAFTAR TABEL

Tabel 2.1 Data <i>Training</i> Deteksi Jatuh.....	21
Tabel 2.2 Data Baru yang akan Diklasifikasi	22
Tabel 2.3 Hasil Penghitungan Jarak Antar Titik	23
Tabel 3.1 Karakteristik Pengguna	27
Tabel 3.2 Kebutuhan Fungsional Aplikasi	30
Tabel 5.1 Hasil Uji Coba Menjalankan Aplikasi Lansia Care.....	88
Tabel 5.2 Hasil Uji Coba <i>Log In</i> pada Aplikasi	89
Tabel 5.3 Hasil Uji Coba <i>Sign Up</i> pada Aplikasi	90
Tabel 5.4 Hasil Uji Coba Mengatur Waktu Minum Obat	93
Tabel 5.5 Hasil Uji Coba Mengatur Jadwal Pemeriksaan Kesehatan	96
Tabel 5.6 Hasil Uji Coba Ubah Profil Pengasuh	100
Tabel 5.7 Hasil Uji Coba Pengaturan Aplikasi.....	102
Tabel 5.8 Hasil Uji Coba Panggilan Darurat dengan <i>Speech Recognition</i>	105
Tabel 5.9 Hasil Uji Coba Memunculkan Notifikasi	106
Tabel 5.10 Hasil Uji Coba Pengiriman Pesan Singkat pada Pengasuh.....	110
Tabel 5.11 Kuisioner Uji Coba Kegunaan Pada Lansia	111
Tabel 5.12 Responden Lansia.....	113
Tabel 5.13 Hasil Kuisioner Uji Coba Kegunaan pada Lansia ...	113
Tabel 5.14 Kuisioner Uji Coba Kegunaan pada Pengasuh Lansia	115
Tabel 5.15 Responden Pengasuh	117
Tabel 5.16 Hasil Kuisioner Uji Coba Kegunaan pada Pengasuh	117
Tabel 5.17 Hasil Uji Coba Deteksi Jatuh pada Aktivitas Pengguna	121
Tabel 5.18 Rangkuman Hasil Pengujian	124

DAFTAR GAMBAR

Gambar 2.1 Koordinat Sistem yang Digunakan oleh API Sensor Android.....	8
Gambar 2.2 <i>Permission</i> untuk <i>Speech Recognition</i>	15
Gambar 2.3 <i>Permission</i> untuk Pengiriman SMS.....	15
Gambar 2.4 <i>Permission</i> untuk Mengakses GPS.....	16
Gambar 3.1 Arsitektur Aplikasi Lansia <i>Care</i>	26
Gambar 3.2 Tahapan Spesifikasi Kebutuhan	29
Gambar 3.3 Diagram Kasus Penggunaan	33
Gambar 3.4 Diagram Alir Teknik <i>Windows Sampling</i> dan <i>Overlapping</i>	36
Gambar 3.5 Diagram Alir Teknik Deteksi Jatuh	37
Gambar 3.6 Diagram Alir Proses Panggilan Darurat	39
Gambar 3.7 Diagram Aktivitas Melakukan Panggilan Darurat...	40
Gambar 3.8 Diagram Aktivitas Melakukan Konfirmasi Deteksi Jatuh.....	41
Gambar 3.9 Diagram Aktivitas Melihat Pengingat Waktu Periksa Kesehatan	42
Gambar 3.10 Aktivitas Melihat Pengingat Waktu Minum Obat	42
Gambar 3.11 Diagram Aktivitas Melakukan Konfirmasi Keadaan Saat Ini.....	43
Gambar 3.12 Diagram Aktivitas Menerima Laporan Lansia Melalui SMS.....	44
Gambar 3.13 Diagram Aktivitas Mengatur Waktu Periksa Kesehatan Lansia.....	44
Gambar 3.14 Mengatur Waktu Minum Obat Lansia.....	45
Gambar 3.15 Diagram Aktivitas Mengatur Durasi Konfirmasi Keadaan.....	46
Gambar 3.16 Rancangan Antarmuka Halaman <i>Sign In</i>	47
Gambar 3.17 Rancangan Antarmuka Halaman <i>Sign Up</i>	48
Gambar 3.18 Rancangan Antarmuka Halaman Utama.....	48
Gambar 3.19 Rancangan Antarmuka Halaman Pengaturan	49
Gambar 3.20 Rancangan Antarmuka Halaman Atur Obat	50
Gambar 3.21 Rancangan Antarmuka Halaman Atur Jadwal	50

Gambar 3.22 Rancangan Antarmuka Halaman Ubah Profil	51
Gambar 3.23 Rancangan Antarmuka Halaman Tambah Waktu Minum Obat	52
Gambar 3.24 Rancangan Antarmuka Halaman Tambah Jadwal Pemeriksaan Kesehatan Lansia	52
Gambar 3.25 Rancangan Antarmuka Notifikasi Waktu Minum Obat	53
Gambar 3.26 Rancangan Antarmuka Notifikasi Jadwal Pemeriksaan Kesehatan	54
Gambar 3.27 Rancangan Antarmuka Notifikasi Konfirmasi Keadaan	54
Gambar 3.28 Rancangan Antarmuka Notifikasi Konfirmasi Jatuh	55
Gambar 4.1 <i>Pseudocode</i> Implementasi Inisialisasi <i>SensorListener</i>	58
Gambar 4.2 <i>Pseudocode</i> Implementasi Mendapatkan Data Sensor <i>Accelerometer</i> dan <i>Gyroscope</i>	59
Gambar 4.3 <i>Pseudocode</i> Implementasi Teknik <i>Windows Sampling</i> dan <i>Overlapping</i>	60
Gambar 4.4 <i>Pseudocode</i> Implementasi Menghitung Rataan Data Sensor	61
Gambar 4.5 <i>Pseudocode</i> Implementasi Menghitung Simpangan Baku Data Sensor	61
Gambar 4.6 <i>Pseudocode</i> Implementasi Membuat <i>Classifier</i>	62
Gambar 4.7 <i>Pseudocode</i> Implementasi Klasifikasi Data Sensor	63
Gambar 4.8 <i>Pseudocode</i> Implementasi Mendapatkan Lokasi Pengguna	64
Gambar 4.9 <i>Pseudocode</i> Implementasi Mendaftarkan Pengingat Minum Obat	65
Gambar 4.10 <i>Pseudocode</i> Mendapatkan Interval <i>Alarm</i> Waktu Minum Obat	65
Gambar 4.11 <i>Pseudocode</i> Implementasi Mendaftarkan Pengingat Pemeriksaan Kesehatan	66
Gambar 4.12 <i>Pseudocode</i> Mendapatkan Interval Waktu Pemeriksaan Kesehatan	67

Gambar 4.13 <i>Pseudocode</i> Implementasi Mengatur Durasi Konfirmasi Keadaan.....	68
Gambar 4.14 <i>Pseudocode</i> Implementasi <i>Text-to-Speech</i> untuk Notifikasi.....	69
Gambar 4.15 <i>Pseudocode</i> Implementasi Melakukan Panggilan Darurat dengan <i>SpeechRecognition</i>	70
Gambar 4.16 <i>Pseudocode</i> Implementasi Pengiriman Pesan Singkat Kepada Pengasuh.....	71
Gambar 4.17 Implementasi Antarmuka Menu Awal.....	72
Gambar 4.18 Implementasi <i>Alert Dialog</i> Kesalahan <i>Sign Up</i>	73
Gambar 4.19 Implementasi Antarmuka Menu <i>Log In</i>	73
Gambar 4.20 Implementasi <i>Alert Dialog</i> Kesalahan <i>Log In</i>	74
Gambar 4.21 Implementasi Antarmuka Menu <i>Sign Up</i>	74
Gambar 4.22 Implementasi Antarmuka Menu Utama.....	75
Gambar 4.23 Implementasi Antarmuka Menu Pengaturan	76
Gambar 4.24 Implementasi <i>Alert Dialog</i> Simpan Pengaturan	76
Gambar 4.25 Implementasi <i>Alert Dialog</i> Pengaturan GPS	77
Gambar 4.26 Implementasi Antarmuka Menu Atur Obat	77
Gambar 4.27 Implementasi Antarmuka Menu Tambah Waktu Minum Obat	78
Gambar 4.28 Implementasi <i>Alert Dialog</i> Penghapusan Data Waktu Minum Obat	78
Gambar 4.29 Implementasi Antarmuka Menu Atur Jadwal	79
Gambar 4.30 Implementasi <i>Alert Dialog</i> Penghapusan Data Jadwal Pemeriksaan.....	80
Gambar 4.31 Implementasi Antarmuka Menu Ubah Profil.....	80
Gambar 4.32 Implementasi Antarmuka Menu Tambah Waktu Minum Obat	81
Gambar 4.33 Implementasi Antarmuka Menu Tambah Waktu Pemeriksaan Kesehatan.....	82
Gambar 4.34 Implementasi Antarmuka Notifikasi Waktu Minum Obat	83
Gambar 4.35 Implementasi Antarmuka Notifikasi Waktu Pemeriksaan Kesehatan	83

Gambar 4.36 Implementasi Antarmuka Notifikasi Konfirmasi Keadaan	84
Gambar 4.37 Implementasi Antarmuka Notifikasi Konfirmasi Terjatuh	85
Gambar 5.1 Grafik Nilai Penghitungan Hasil Uji Coba	123
Gambar B. 1 Gambar Hasil Menu Awal Aplikasi	137
Gambar B. 2 Gambar Hasil Masuk Menu <i>Log In</i>	137
Gambar B. 3 Gambar Hasil Masuk Menu Utama Aplikasi	138
Gambar B. 4 Gambar Hasil <i>Alert Dialog</i> Kesalahan <i>Login</i>	138
Gambar B. 5 Gambar Hasil Masuk Menu <i>Sign Up</i>	139
Gambar B. 6 Gambar Hasil <i>Alert Dialog</i> Kesalahan Ketidakcocokan	139
Gambar B. 7 Gambar Hasil <i>Alert Dialog</i> Kesalahan Data Tidak Lengkap	140
Gambar B. 8 Gambar Hasil Masuk Menu Atur Waktu Minum Obat	140
Gambar B. 9 Gambar Hasil Menu Tambah Obat	141
Gambar B. 10 Gambar Hasil Menampilkan Daftar Waktu Minum Obat	141
Gambar B. 11 Gambar Hasil Menu Ubah Waktu Minum Obat	142
Gambar B. 12 Gambar Hasil <i>Alert Dialog</i> Hapus Waktu Minum Obat	142
Gambar B. 13 Gambar Hasil Menu Atur Jadwal Pemeriksaan Kesehatan	143
Gambar B. 14 Gambar Hasil Menu Tambah Waktu Pemeriksaan Kesehatan	143
Gambar B. 15 Gambar Hasil Menampilkan Daftar Waktu Pemeriksaan Kesehatan	144
Gambar B. 16 Gambar Hasil Menu Ubah Waktu Pemeriksaan Kesehatan	144
Gambar B. 17 Gambar Hasil <i>Alert Dialog</i> Hapus Waktu Pemeriksaan Kesehatan	145
Gambar B. 18 Gambar Hasil Menu Ubah Profil Pengasuh	145

Gambar B. 19 Gambar Hasil <i>Alert Dialog</i> Kesalahan <i>Password</i> dan <i>Confirm Password</i> Tidak Cocok	146
Gambar B. 20 Gambar Hasil <i>Alert Dialog</i> Kesalahan Data Pengasuh Tidak Lengkap	146
Gambar B. 21 Gambar Hasil Menu Pengaturan	147
Gambar B. 22 Gambar Hasil Mengatur Durasi dan Nomor <i>Handphone</i>	147
Gambar B. 23 Gambar Hasil Menekan Pilihan Simpan	148
Gambar B. 24 Gambar Hasil <i>Error Text</i> Keharusan Pengisian Nomor <i>Handphone</i>	148
Gambar B. 25 Gambar Hasil Melakukan Panggilan Darurat Kepada Pengasuh.....	149
Gambar B. 26 Gambar Hasil Notifikasi Waktu Minum Obat ...	149
Gambar B. 27 Gambar Hasil Notifikasi Waktu Pemeriksaan Kesehatan	150
Gambar B. 28 Gambar Hasil Notifikasi Konfirmasi Keadaan ..	150
Gambar B. 29 Gambar Hasil Notifikasi Deteksi Jatuh.....	151
Gambar B. 30 Gambar Hasil Pesan Singkat pada <i>Handphone</i> Pengasuh.....	151

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, dan sistematika penulisan.

1.1 Latar Belakang Masalah

Seiring meningkatnya derajat kesehatan dan kesejahteraan penduduk akan berpengaruh pada peningkatan Usia Harapan Hidup (UHH) di Indonesia. Dengan semakin meningkatnya usia harapan hidup penduduk, menyebabkan jumlah penduduk lanjut usia (lansia) terus meningkat dari tahun ke tahun. Menurut Undang-undang Nomor 13 tahun 1998 tentang Kesejahteraan Lansia, yang dimaksud dengan lansia adalah penduduk yang telah mencapai usia 60 tahun ke atas. Hasil sensus menunjukkan bahwa penduduk lansia Indonesia pada tahun 2013 mencapai 21 juta atau 9,6% dari keseluruhan penduduk Indonesia [1]. Secara global, kebanyakan lansia hidup mandiri, apakah dengan status hidup sendiri atau dengan pasangannya. Hanya $\pm 5\%$ lansia yang hidup di panti. Sekitar 65% lansia yang mengalami gangguan kesehatan hidup ditemani oleh seseorang yang mengingat masalah kesehatannya. Adapun selebihnya, sebesar 35%, memang hidup sendiri [2].

Lansia yang tinggal sendiri rentan mengalami masalah baik fisik maupun psikologis. Masalah-masalah tersebut diantaranya adalah rentannya lansia untuk jatuh, ketakutan akan kematian, pengelihan yang berkurang, kepikunan, dan rasa terisolasi [3]. Masalah tersebut dapat berkembang menjadi keadaan darurat apabila tidak segera ditangani. Oleh karena itu, dibutuhkan sebuah aplikasi yang dapat menghubungkan lansia dengan pengasuhnya, baik itu pengasuh maupun anaknya, untuk mengetahui keadaan lansia sehingga lansia dapat dengan cepat mendapatkan pertolongan dalam keadaan darurat.

Untuk mengatasi masalah tersebut, akan dibangun sebuah aplikasi Android yang menjembatani hubungan antara pengasuh dengan lansia. Aplikasi Android ini dapat mendeteksi apakah lansia terjatuh dengan menggunakan data sensor *accelerometer* dan *gyroscope* pada perangkat keras Android dan mengirimkan pesan darurat kepada pengasuh melalui SMS. Selain itu, aplikasi ini juga memfasilitasi lansia untuk melaporkan status darurat dengan menggunakan perintah suara (*Speech recognition*) kepada pengasuh melalui SMS, dan menggunakan *text-to-speech* untuk mengingatkan lansia waktu pemeriksaan kesehatan dan waktu minum obat serta untuk menanyakan keadaan lansia pada waktu yang telah ditentukan dan mengirimkan laporannya kepada pengasuh melalui SMS.

Implementasi “Rancang Bangun Aplikasi Lansia *Care* untuk *Monitoring* Lansia yang Tinggal Sendiri Berbasis Android” ini diharapkan mampu menjadi penghubung antara lansia yang tinggal sendiri dengan pengasuhnya secara tepat dan cepat.

1.2 Rumusan Permasalahan

Rincian permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara mendeteksi pengguna terjatuh menggunakan sensor *accelerometer* dan *gyroscope* yang telah disediakan perangkat keras Android?
2. Bagaimana cara menerapkan *Speech Recognition* untuk mengenali kata yang diucapkan pengguna merupakan kalimat permintaan tolong?
3. Bagaimana cara menerapkan *Text-to-Speech* untuk memperingatkan lansia dalam waktu pemeriksaan kesehatan dan waktu minum obat?
4. Bagaimana cara menerapkan *Text-to-Speech* untuk notifikasi konfirmasi keadaan pada waktu yang ditentukan?

5. Bagaimana cara mengambil koordinat lokasi pengguna saat ini untuk dikirimkan kepada pengasuh lansia saat keadaan darurat?
6. Bagaimana cara mengirimkan pesan SMS melalui aplikasi Android?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, antara lain:

1. Aplikasi berbasis Android.
2. *Smartphone* diletakkan pada saku celana sebelah kiri pengguna.
3. Bahasa pemrograman yang digunakan adalah Java.
4. Penulis menggunakan API yang telah disediakan oleh Android.
5. Penulis menggunakan sensor *accelerometer* dan *gyroscope* yang telah disediakan dalam perangkat keras Android.
6. Laporan keadaan lansia dikirimkan dalam bentuk pesan SMS.
7. Aplikasi *Lansia Care* dapat mengurangi daya tahan baterai *smartphone* karena mengaktifkan *service* secara kontinyu.

1.4 Tujuan

Tujuan dalam pembuatan tugas akhir ini antara lain:

1. Membuat aplikasi Android untuk membantu lansia dalam mengingat jadwal pemeriksaan kesehatan dan waktu minum obat.
2. Membuat aplikasi Android untuk memberitahukan bahwa lansia sedang dalam keadaan darurat.
3. Membuat aplikasi Android yang dapat mendeteksi bahwa lansia terjatuh.
4. Membuat aplikasi Android yang dapat melaporkan keadaan lansia, dengan *trigger-trigger* yang telah ditentukan, kepada pengasuh.

1.5 Manfaat

Manfaat pembuatan tugas akhir ini antara lain:

1. Menghubungkan lansia yang tinggal sendiri dengan pengasuhnya.
2. Membantu lansia untuk mengatasi masalah kepikunan, seringnya terjatuh, ketakutan, rasa terisolasi, dan kurangnya pengelihatan.

1.6 Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, permasalahan, batasan masalah, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

BAB III. PERANCANGAN PERANGKAT LUNAK

Bab ini membahas tahap analisis permasalahan dan perancangan dari aplikasi yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir.

BAB IV. IMPLEMENTASI

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Bab ini berisi proses implementasi dari setiap fungsi pada aplikasi.

BAB V. UJI COBA DAN EVALUASI

Bab ini membahas hasil-hasil uji coba yang telah dilakukan. Bab ini berisi hasil uji coba fungsionalitas, uji coba kegunaan, dan uji coba akurasi.

BAB VI. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya

BAB II TINJAUAN PUSTAKA

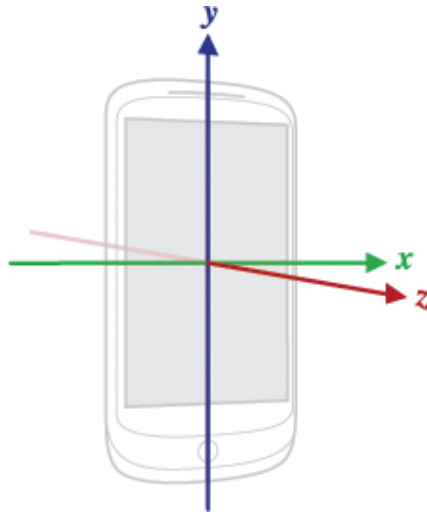
Bab ini berisi penjelasan teori-teori yang berkaitan dengan metode yang diajukan pada pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Sensor

Sensor merupakan sebuah perangkat yang mendeteksi perubahan dari stimulus fisik dan mengubahnya menjadi sinyal yang dapat diukur maupun direkam [4]. Sebagian besar perangkat Android memiliki *built-in* sensor yang mengukur gerak, orientasi, dan berbagai kondisi lingkungan [5]. Sensor tersebut mampu memberikan data mentah dengan presisi tinggi dan akurat, dan berguna jika pengembang ingin memantau pergerakan perangkat dalam tiga dimensi, atau untuk memonitor perubahan pada lingkungan sekitar perangkat. *Platform* Android mendukung tiga kategori sensor, yaitu:

- *Motion Sensor*
Sensor ini mengukur percepatan dan rotasi dalam tiga sumbu. Kategori ini meliputi *accelerometer*, *gravity sensors*, *gyroscope*, dan *rotational vector sensor*.
- *Environmental Sensor*
Sensor ini menghitung beberapa parameter lingkungan, seperti suhu udara dan tekanan, pencahayaan, dan kelembaban. Kategori ini meliputi *barometer*, *photometer*, dan *thermometer*.
- *Position Sensor*
Sensor ini mengukur posisi fisik dari perangkat. Kategori ini meliputi *orientation sensor* dan *magnetometer*.

Gambar 2.1 merupakan gambar koordinat sensor yang digunakan oleh API Sensor Android.



Gambar 2.1 Koordinat Sistem yang Digunakan oleh API Sensor Android

2.1.1. Sensor Accelerometer

Accelerometer adalah elemen *sensing* yang mengukur percepatan. Percepatan adalah laju perubahan kecepatan terhadap waktu. *Accelerometer* mengukur unit dalam satuan g dimana g adalah pengukuran percepatan gravitasi yang sama dengan $9,82 \text{ m/s}^2$. *Accelerometer* berkembang dari tabung air sederhana dengan gelembung udara yang menunjukkan arah percepatan ke sirkuit terpadu yang dapat ditempatkan pada papan sirkuit. *Accelerometer* dapat mengukur getaran, guncangan, kemiringan, dampak dan gerak dari suatu objek [6]. *Accelerometer* pada perangkat bergerak digunakan untuk menghitung percepatan linear dari gerakan. *Accelerometer* akan mengukur arah gerakan perangkat [7].

2.1.2. Sensor Gyroscope

Gyroscope adalah alat untuk mengukur atau mempertahankan orientasi, berdasarkan prinsip-prinsip

momentum sudut [8]. *Gyroscope* pada perangkat bergerak digunakan untuk menghitung kecepatan rotasi sudut. *Gyroscope* akan menghitung orientasi lateral atau kemiringan(*tilt*) dari gerakan perangkat [7].

2.2 Activity Recognition

Activity recognition merupakan teknik yang digunakan untuk proses pendeteksian aktivitas fisik pengguna [9]. Adapun tahap-tahap dari *activity recognition* dapat dijelaskan sebagai berikut:

2.2.1. Data Collection

Untuk melakukan sebuah identifikasi aktivitas dikatakan jatuh atau tidak, diperlukan data yang nantinya akan digunakan untuk pengolahan. Pada umumnya data diambil dengan menggunakan sebuah perangkat berupa sensor. Data ini diharapkan dapat digunakan untuk dilakukan proses identifikasi yang bertujuan untuk menggambarkan sebuah entitas tertentu.

2.2.2. Feature Extraction

Setelah memperoleh data dari sensor *accelerometer* dan *gyroscope*, proses selanjutnya adalah ekstraksi data. Data yang akan diproses terlebih dahulu diolah sedemikian hingga agar data tersebut tidak terlalu konvergen atau terlalu divergen untuk dilakukan proses pendeteksian menggunakan klasifikasi *k-Nearest Neighbour*. Salah satunya menggunakan teknik *windows sampling* dan *overlapping*.

2.2.3. Data Interpretation

Hal ini merupakan tahap paling penting dalam proses identifikasi aktivitas pengguna. Setelah memiliki data yang sudah diekstraksi maka langkah selanjutnya adalah melakukan

pendeteksian dengan menggunakan klasifikasi *k-Nearest Neighbour*.

2.3 Sistem Operasi Android

Sistem Operasi Android adalah sistem operasi untuk alat dengan daya kecil yang bekerja dengan menggunakan baterai dan penuh dengan perangkat keras seperti penerima *Global Positioning System (GPS)*, kamera, sensor cahaya dan orientasi, WiFi, konektifitas *telephony*, dan layar sentuh. Seperti operasi sistem lainnya, Android mengizinkan aplikasi untuk menggunakan perangkat keras melalui abstraksi dan menyediakan lingkungan untuk aplikasi. *Platform* Android dibangun oleh Android Inc. yang kemudian dibeli oleh Google dan dirilis sebagai *Android Open Source Project (AOSP)* pada tahun 2007 [10].

2.3.1. Komponen Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman *Java*. *Android Software Development Kit (SDK)* akan mengkompilasi kode beserta dengan data dan berkas menjadi APK, yang merupakan paket Android berupa berkas arsip dengan ekstensi .apk. Satu APK mengandung semua komponen dari aplikasi Android dan merupakan berkas untuk instalasi aplikasi. Komponen aplikasi adalah bagian penting dari sebuah aplikasi Android. Terdapat empat jenis komponen aplikasi yang berbeda. Tiap jenis memiliki tujuan berbeda dan memiliki siklus hidup berbeda yang mendefinisikan bagaimana komponen dibuat dan dihancurkan. Empat jenis komponen aplikasi Android adalah *activity*, *service*, *content provider*, dan *broadcast receiver* [11].

2.3.2. Activity

Sebuah *activity* merepresentasikan satu layar dengan antarmuka pengguna. Sebagai contoh, sebuah aplikasi pesan elektronik memiliki satu *activity* untuk menampilkan daftar pesan

baru, *activity* lain untuk menulis pesan, dan *activity* untuk membaca pesan. Walaupun *activity* tersebut bekerja bersamaan, setiap *activity* tidak saling bergantung.

2.3.3. *Service*

Sebuah *service* merupakan komponen yang berjalan di *background* untuk melakukan operasi yang berjalan terus-menerus atau untuk melakukan pekerjaan untuk proses *remote*. Sebuah *service* tidak membutuhkan antarmuka pengguna. Sebagai contoh, sebuah *service* dapat memainkan musik pada *background* ketika pengguna menjalankan aplikasi lain tanpa menghalangi interaksi pengguna dengan *activity*. Komponen lain seperti *activity* dapat memulai *service* dengan langsung menjalankannya atau dengan melakukan *bind* pada *service* untuk dapat saling berinteraksi.

2.3.4. *Content Provider*

Sebuah *content provider* mengelola seperangkat data aplikasi. Data dapat disimpan dalam sistem, SQLite *database*, *web*, maupun lokasi penyimpanan persisten lainnya yang dapat diakses oleh aplikasi. Aplikasi lain dapat melakukan *query* atau bahkan modifikasi data melalui *content provider*. Sebagai contoh, Android menyediakan *content provider* untuk mengelola informasi kontak pengguna, dengan demikian aplikasi yang memiliki izin dapat melakukan *query* pada data kontak pengguna.

2.3.5. *Broadcast Receiver*

Sebuah *broadcast receiver* merupakan komponen yang merespon pengumuman *broadcast* ke seluruh sistem. Terdapat banyak *broadcast* yang berasal dari sistem, contohnya *broadcast* yang mengumumkan layar telah dimatikan, baterai rendah, dan lain-lain. Aplikasi juga dapat memulai *broadcast*, sebagai contoh untuk memberitahu aplikasi lain bahwa data telah diunduh dan siap untuk digunakan. Pada umumnya, *broadcast receiver* hanya

bertindak sebagai gerbang untuk komponen lain dan dimaksudkan untuk melakukan jumlah pekerjaan yang sangat sedikit.

2.3.6. *SensorEventListener*

SensorEventListener merupakan *public interface* yang digunakan untuk menerima notifikasi dari *SensorManager* ketika nilai sensor berubah. *SensorEventListener* memiliki dua *public method* yaitu *onAccuracyChanged* dan *onSensorChanged*. Metode *onAccuracyChanged* dipanggil ketika akurasi dari sensor mengalami perubahan. Metode *onSensorChanged* dipanggil ketika nilai sensor berubah [12].

2.3.7. *RecognitionListener*

RecognitionListener merupakan *public interface* yang digunakan untuk menerima notifikasi dari *SpeechRecognizer* ketika *recognition event* terjadi [13]. Terdapat sembilan *public method* sebagai berikut:

- *onBeginningOfSpeech*
Dipanggil ketika pengguna mulai berbicara.
- *onBufferReceived*
Dipanggil ketika suara lain diterima. Tujuan dari fungsi ini adalah untuk memberikan umpan balik kepada pengguna berkaitan dengan suara yang ditangkap.
- *onEndOfSpeech*
Dipanggil setelah pengguna berhenti berbicara.
- *onError*
Dipanggil ketika terjadi kesalahan jaringan ataupun penangkapan suara.
- *onEvent*
Disediakan untuk menambahkan *future event*.
- *onPartialResults*
Dipanggil ketika hasil *partial recognition* tersedia.

- *onReadyForSpeech*
Dipanggil ketika *endpointer* siap untuk menangkap suara pengguna.
- *onResults*
Dipanggil ketika hasil penangkapan suara siap.
- *onRmsChanged*
Dipanggil ketika level suara pada *audio stream* berubah.

2.3.8. *LocationListener*

LocationListener digunakan untuk menghubungkan aplikasi Android dengan perangkat GPS bersama dengan *LocationManager* [14]. *LocationListener* menerima notifikasi dari *LocationManager* ketika lokasi pengguna berubah. Metode ini dipanggil apabila *LocationListener* telah diregistrasi dengan *location manager service* menggunakan metode *requestLocationUpdates*. Terdapat empat *public method* sebagai berikut:

- *onLocationChanged*
Dipanggil ketika lokasi berubah.
- *onProviderDisabled*
Dipanggil ketika provider dinonaktifkan oleh pengguna.
- *onProviderEnabled*
Dipanggil ketika provider diaktifkan oleh pengguna.
- *onStatusChanged*
Dipanggil ketika status provider berubah.

2.3.9. *SmsManager*

SmsManager merupakan *public final class* yang mengelola operasi SMS seperti mengirim data, teks, dan pesan pdu SMS [15]. Untuk mendapatkan objek ini dilakukan pemanggilan *static*

method getDefault. Terdapat delapan *public method* sebagai berikut:

- *divideMessage*
Digunakan untuk membagi pesan teks menjadi beberapa fragmen yang tidak lebih besar dari ukuran pesan SMS maksimal.
- *downloadMultimediaMessage*
Digunakan untuk mengunduh pesan MMS dari *carrier* dengan lokasi URL yang telah diberikan.
- *getCarrierConfigValues*
Digunakan untuk mendapatkan nilai konfigurasi *carrier-dependent*.
- *getDefault*
Digunakan untuk mendapatkan objek *SmsManager* yang berkaitan dengan id *subscription*.
- *sendDataMessage*
Digunakan untuk mengirim data SMS ke *port* aplikasi tertentu.
- *sendMultimediaMessage*
Digunakan untuk mengirim pesan MMS.
- *sendMultipartTextMessage*
Digunakan untuk mengirim teks SMS dengan banyak bagian.
- *sendTextMessage*
Digunakan untuk mengirim pesan SMS berbasis teks.

2.4 Text to Speech

Text to Speech adalah sistem untuk mengubah teks yang diberikan menjadi gelombang suara. Konversi melibatkan pengolahan teks dan proses menghasilkan suara. Proses ini memiliki koneksi ke teori linguistik, model produksi suara, dan karakterisasi akustik-fonetik bahasa [16].

2.5 *Speech Recognition*

Speech Recognition atau pengenalan suara komputer adalah proses mengonversi sinyal suara ke urutan kata-kata melalui sebuah algoritma yang diimplementasikan pada program komputer [17]. Dibutuhkan *permission* yang digunakan agar aplikasi Android dapat menggunakan fitur *speech recognition* dengan menambahkan *permission* seperti pada Gambar 2.2.

```
<uses-permission  
Android:name="Android.permission.RECORD_AUDIO" />
```

Gambar 2.2 *Permission* untuk *Speech Recognition*

2.6 *Short Message Service (SMS)*

Short Message Service (SMS) menyediakan sarana untuk mengirim pesan dengan ukuran terbatas, 160 karakter, ke dan dari peralatan terminal. SMS pada awalnya distandarisasi dan diimplementasikan pada jaringan GSM, namun pengiriman SMS dengan *fixed network* juga telah diperkenalkan [18].

Pesan SMS ditangani oleh *short message service center (SMSC)* yang dikelola oleh provider seluler. Ketika pengguna mengirim pesan SMS kepada pengguna lain, pesan tersebut dikirimkan ke SMSC untuk menyimpan pesan dan mengirimkannya apabila penerima berada dalam jaringan. Dibutuhkan *permission* yang digunakan agar aplikasi Android dapat menggunakan fitur pengiriman SMS dengan menambahkan *permission* seperti pada Gambar 2.3.

```
<uses-permission  
Android:name="Android.permission.SEND_SMS" />
```

Gambar 2.3 *Permission* untuk Pengiriman SMS

2.7 *Global Positioning System (GPS)*

Global Positioning System (GPS) adalah sistem radionavigasi ruang angkasa yang dikelola oleh *US Air Force*

(USAF) sebagai operator sistem. GPS pada awalnya dikembangkan untuk sistem kekuatan tambahan bagi militer [19]. Namun, GPS juga memiliki potensi menguntungkan bagi masyarakat sipil dalam berbagai aplikasi. Dalam rangka membuat GPS menjadi layanan yang bermanfaat bagi banyak pengguna bersamaan dengan memastikan bahwa kepentingan keamanan nasional Amerika Serikat terus diamati, GPS menyediakan dua layanan. Layanan pertama adalah *Precise Positioning Service (PPS)* yang tersedia untuk kekuatan militer AS dan sekutunya, yang kedua adalah *Standard Positioning Service (SPS)* yang didesain untuk memberikan kemampuan *positioning* untuk dimanfaatkan masyarakat sipil diseluruh dunia .

Dibutuhkan *permission* yang digunakan agar aplikasi Android dapat menggunakan fitur GPS pada *hardware* Android dengan menambahkan *permission* seperti pada Gambar 2.4.

```
<uses-permission
Android:name="Android.permission.ACCESS_FINE_LOCATION" />
```

Gambar 2.4 *Permission* untuk Mengakses GPS

2.8 *Training Set*

Training set adalah seperangkat data yang digunakan dalam berbagai bidang ilmu informasi untuk menemukan hubungan potensial yang diprediksikan. *Training set* digunakan dalam kecerdasan buatan, *machine learning*, *genetic programming*, sistem cerdas, dan statistika [20].

2.8.1. *Data Training*

Data *training* berisi rata-rata hasil dari pembacaan sensor *accelerometer* serta *gyroscope* dalam sumbu *x*, *y*, *z*, serta simpangan baku dari masing-masing sumbu sensor *accelerometer* dan *gyroscope* dan kelas klasifikasi.

2.8.2. Rata-Rata

Dalam bahasa sehari-hari, rata-rata adalah jumlah dari sebuah daftar dibagi dengan banyak data pada daftar [21]. Dalam matematika dan statistika, rata-rata disebut dengan *arithmetic mean*. Selain itu, rata-rata juga dapat merujuk pada median, mode, ataupun nilai pusat atau tipikal. Dalam statistika, dikenal sebagai ukuran tendensi sentral. Persamaan 2.1 berikut adalah persamaan untuk mencari rata-rata.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.1)$$

Keterangan:

\bar{x} = rata-rata data

n = jumlah data

x_i = data pada indeks ke i

2.8.3. Simpangan Baku

Dalam statistika dan probabilitas, simpangan baku atau deviasi standar adalah ukuran sebaran statistik yang paling lazim [22]. Singkatnya, simpangan baku mengukur bagaimana nilai-nilai data tersebar. Bisa juga didefinisikan sebagai rata-rata jarak penyimpangan titik-titik data diukur dari nilai rata-rata data tersebut. Persamaan 2.2 berikut adalah persamaan untuk mencari simpangan baku.

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (2.2)$$

Keterangan:

S = simpangan baku

n = jumlah data

x_i = data pada indeks ke i

\bar{x} = rata-rata data

2.9 *Java Machine Learning (Java-ML)*

Java-ML adalah sekumpulan *machine learning* dan algoritma *data mining* yang bertujuan untuk menjadi API yang mudah digunakan dan mudah dikembangkan untuk pengembang perangkat lunak maupun para peneliti [23]. Antarmuka untuk setiap jenis algoritma dibuat sangat sederhana dan algoritma mengikuti antarmuka masing-masing secara ketat. Membandingkan *classifier* yang berbeda atau algoritma *clustering* sangat mudah, dan pengimplementasian algoritma baru juga sangat mudah. Implementasi dari algoritma ditulis dengan jelas, didokumentasikan dengan lengkap, dan dapat digunakan sebagai referensi. *Library* Java-ML ditulis dalam bahasa pemrograman Java dan tersedia dari <http://java-ml.sourceforge.net/> di bawah lisensi GNU GPL.

2.10 **Klasifikasi**

Klasifikasi memiliki dua makna yang berbeda. Pada [24] diberikan sebuah set pengamatan dengan tujuan menetapkan keberadaan kelas atau kluster dalam data. Atau sudah diketahui secara pasti dalam data terdapat banyak kelas, dan tujuannya adalah untuk menentukan aturan dimana data observasi dapat diklasifikasikan menjadi salah satu dari kelas yang sudah ada. Jenis sebelumnya dikenal sebagai *unsupervised learning* (atau *clustering*), setelahnya dikenal sebagai *supervised learning*. Pada [24] istilah klasifikasi berarti *supervised learning*. Dalam literatur statistik, *supervised learning* biasanya disebut sebagai *discrimination*, yang berarti penetapan aturan klasifikasi dari data yang telah diklasifikasikan atau dikelompokkan dengan benar.

2.11 *k-Nearest Neighbour*

k-Nearest Neighbour merupakan salah satu algoritma klasifikasi yang bertujuan untuk menggunakan *dataset* dimana

setiap titik datanya akan dipisah menjadi beberapa kelas untuk memprediksi klasifikasi dari sebuah titik data baru [25].

Secara umum, dimulai dengan satu set data, dimana setiap datanya sudah diketahui kelasnya. Kemudian, memprediksi klasifikasi dari data baru berdasarkan observasi pada set data. Proses pemilihan klasifikasi dari observasi baru disebut dengan *classification problem*. *k-Nearest Neighbour* melakukan pemilihan klasifikasi dari data baru berdasarkan observasi pada set data yang paling mirip.

Setiap karakteristik dari set data dianggap sebagai dimensi berbeda pada sebuah *space*, dan mengambil nilai pengamatan dari karakteristik untuk dijadikan koordinat pada dimensi, sehingga didapatkan serangkaian titik pada *space*. Kemudian dapat dipertimbangkan kesamaan antar dua titik berdasarkan jarak antar keduanya dengan menggunakan metrik yang sesuai.

Cara algoritma ini menentukan mana titik pada *training set* yang memiliki cukup kesamaan dengan data observasi baru untuk dapat dipilih kelas prediksinya adalah dengan memilih k titik yang paling dekat dengan titik data observasi, dan mengambil kelas yang paling umum dari keduanya. Oleh karena itu, algoritma ini disebut dengan *k-Nearest Neighbour*.

Algoritma *k-Nearest Neighbour* adalah sebagai berikut:

1. Menentukan *integer* positif k , bersama dengan sampel baru.
2. Memilih entri k pada set data yang paling dekat dengan sampel baru.
3. Menemukan klasifikasi yang paling umum dari entri.
4. Menentukan klasifikasi pada sampel baru.

Contoh pengolahan data *training* deteksi jatuh dengan klasifikasi *k-Nearest Neighbour* ditampilkan dalam Tabel 2.3, untuk data training secara lengkap terdapat pada Lampiran A. Data *training* yang digunakan terdapat pada Tabel 2.1. Sedangkan, data baru yang akan diklasifikasi terdapat pada Tabel 2.2.

Dari setiap fitur yang terdapat pada data training akan dicari jarak dengan data baru kemudian akan dipilih titik yang paling

dekat dengan titik data baru tersebut. Kelas klasifikasi dapat diambil dari kelas titik yang paling dekat dengan titik data.

Persamaan 2.3 merupakan rumus untuk mencari jarak tiap titik dalam *k-Nearest Neighbour* menggunakan *Euclidean distance*.

$$d = \sqrt{\sum_{i=1}^f (x_i - y_i)^2} \quad (2.3)$$

Keterangan:

d = jarak antara titik x dan y dalam *space*

f = jumlah *feature*

x_i = titik pada *space*

y_i = titik observasi

Tabel 2.1 Data Training Deteksi Jatuh

AccX	AccY	AccZ	StDev AX	StDev AY	StDev AZ	GyroX	GyroY	GyroZ	StDev GX	StDev GY	StDev GZ	Clas s
1.45328	- 3.2901 4.97563	- 3.2901 1	3.3613	4.7535 62	4.6927 7	0.0492 2	0.4869 82	0.1938 12	0.58479 6	0.849 865	0.4252 99	jatuh
2.5886108 55	4.0375817 66	- 2.9214	4.1726 85	5.6665 45	4.1279 05	0.2813 34	0.1820 84	0.1877 19	0.79411 8	0.443 488	0.5478 93	jatuh
2.876873	3.4050336 97	- 4.6720 5	3.7236 28	5.1900 79	5.3913 5	0.4875 62	0.1214 25	0.4215 58	1.91838 8	0.813 014	1.7040 01	jatuh
1.521755	4.5303083 84	- 3.1325 7	4.2298 59	5.6558 6	6.1059 47	0.4776 2	- 0.1109 9	- 0.5655 24	1.58795 6	1.509 722	1.8453 96	jatuh
2.05614	4.3210552 32	- 3.0947 5	4.2831 06	4.9035 38	4.4488 42	0.1559 23	0.2583 04	0.1523 19	0.69169 2	0.570 567	0.4735 69	jatuh
-1.97617	4.9497725 55	4.7160 99	2.1085 79	5.8098 84	5.2350 78	- 0.0413 4	- 0.0347 4	- 0.0502 4	1.44081 8	0.603 513	0.6399 95	tidak jatuh
-2.35973	2.7064055 5	5.6843 14	2.4511 14	6.6295 31	5.3293 68	0.0536 8	- 0.0959 2	- 0.1028 5	1.43338 7	0.852 731	0.8286 52	tidak jatuh
-2.75333	2.9654581 96	6.0827 09	2.4481 82	7.2528	5.5457 85	- 0.1843 1	- 0.2533 7	- 0.3499 6	1.47929 1	0.841 119	0.8844 58	tidak jatuh
-0.62584	9.399157	2.0614 08	1.0397 89	1.3723 04	2.0102 7	0.0061 7	0.3716 2	0.0767 09	0.71191 6	0.539 076	0.3210 29	tidak jatuh

-0.82696	9.7654698 85	1.5863 98	1.0635 42	1.5582 36	1.8111 17	- 0.0525 3	0.7898 03	0.0556 5	0.46975 4	1.028 036	0.1849 27	tidak jatuh
----------	-----------------	--------------	--------------	--------------	--------------	------------------	--------------	-------------	--------------	--------------	--------------	----------------

Tabel 2.2 Data Baru yang akan Diklasifikasi

AccX	AccY	AccZ	StDevA X	StDevA Y	StDev AZ	GyroX	GyroY	GyroZ	StDev GX	StDevG Y	StDevG Z	Cl ass
2.04847 8994	4.79702 2688	- 3.88052 2159	3.31966 6019	5.26473 7601	5.5353 8794	- 0.38452 4406	0.00633 7723	0.48598 9129	2.3515 1717	2.39598 6275	1.90568 9428	Jat uh

Tabel 2.3 Hasil Penghitungan Jarak Antar Titik

Data Training	Kelas	Distance
Data 1	Jatuh	3.148913509
Data 2	Jatuh	3.647337403
Data 3	Jatuh	2.636644699
Data 4	Jatuh	2.085780256
Data 5	Jatuh	3.418824742
Data 6	Tidak Jatuh	9.90093322
Data 7	Tidak Jatuh	11.08523093
Data 8	Tidak Jatuh	11.63690814
Data 9	Tidak Jatuh	10.27203261
Data 10	Tidak Jatuh	10.23825349

Dari Tabel 2.3 diperoleh bahwa jarak terdekat titik data baru adalah dengan data *training* ke 4 yaitu 2.085780256. Sehingga, dapat disimpulkan data baru diklasifikasikan sebagai aktivitas jatuh.

BAB III

PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan hal-hal yang berkaitan dengan perancangan sistem yang akan dibuat dalam Tugas Akhir ini, dimulai dari deskripsi umum mengenai perangkat lunak yang akan dibuat, perancangan proses-proses yang ada, dan arsitektur umum sistem.

3.1 Deskripsi Umum

Dalam tugas akhir ini dibangun sebuah perangkat lunak sistem monitoring lansia yang tinggal sendiri. Sistem ini berfungsi untuk menghubungkan lansia dengan pengasuhnya. Terdapat beberapa fitur yang dapat dimanfaatkan pengasuh untuk memonitor lansia diantaranya adalah pengingat minum obat dan jadwal pemeriksaan, konfirmasi keadaan lansia, panggilan darurat, serta pendeteksi jatuh.

Data *realtime sensor accelerometer* dan *gyroscope* yang ada pada *hardware* Android akan digunakan untuk mendeteksi apakah lansia terjatuh atau tidak yang diolah menggunakan *classifier k-Nearest Neighbour*. Untuk memperoleh hasil yang baik, maka *smartphone* diletakkan pada tempat yang telah dilakukan uji coba *data training*, dalam hal ini *smartphone* diletakkan di saku celana sebelah kiri.

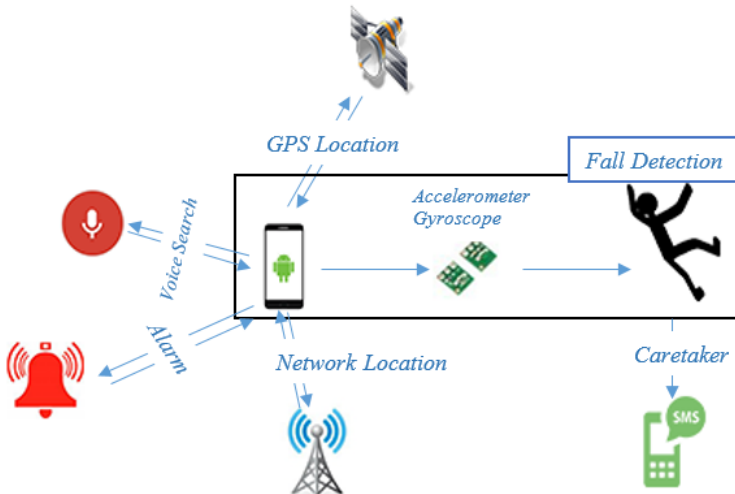
3.2 Arsitektur Sistem

Dalam sistem ini setiap komponen memiliki peran masing-masing dalam sistem. Komponen sistem terdiri sebagai berikut:

- Pengguna
Pengguna yang dimaksud adalah lansia yang menggunakan *smartphone* Android.

- Pengasuh lansia
Pengasuh lansia merupakan orang yang diberikan tanggung jawab untuk melakukan pengaturan aplikasi dan penerima notifikasi.
- *Device*
Device yang digunakan adalah *smartphone* berbasis Android minimal versi 4.0 dan *smartphone* yang telah tertanam sensor *accelerometer* dan *gyroscope*.
- *Sensor Accelerometer*
Sensor *accelerometer* merupakan *hardware* yang melekat pada *smartphone* berbasis Android.
- *Sensor Gyroscope*
Sensor *gyroscope* merupakan *hardware* yang melekat pada *smartphone* berbasis Android.

Adapun perancangan arsitektur sistem yang telah dibuat dapat dilihat pada Gambar 3.1.



Gambar 3.1 Arsitektur Aplikasi Lansia Care

3.3 Karakteristik Pengguna

Terdapat dua pengguna aplikasi Lansia *Care* yaitu:

1. Lansia

Lansia merupakan penduduk yang telah mencapai usia 60 tahun ke atas. Lansia dalam konteks aplikasi merupakan lansia yang tinggal sendiri tanpa didampingi keluarga maupun pengasuhnya, namun lansia masih tergolong lansia mandiri atau tidak dalam keadaan *bedrest*. Aplikasi Lansia *Care* akan di-*install* di *smartphone* Android Lansia untuk melakukan deteksi maupun memberikan notifikasi.

2. Pengasuh Lansia

Pengasuh merupakan aktor yang diberikan tanggung jawab untuk melakukan pengaturan aplikasi dan penerima notifikasi. Pengasuh menerima notifikasi dalam bentuk pesan singkat yang dapat dibuka dengan aplikasi bantuan lainnya. Pengasuh diberikan hak akses untuk masuk ke aplikasi dan melakukan pengaturan.

Karakteristik pengguna dapat dijelaskan pada Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

Kategori Pengguna	Tugas	Hak Akses ke Aplikasi	Kemampuan yang Harus Dimiliki
Lansia	Melakukan panggilan darurat	-	Dapat mengoperasikan <i>smartphone</i> Android secara dasar
	Melakukan konfirmasi deteksi jatuh	-	
	Melihat pengingat	-	

	waktu periksa kesehatan		
	Melihat peringat waktu minum obat	-	
	Melakukan konfirmasi keadaan saat ini	-	
Pengasuh Lansia	Menerima laporan dari lansia melalui SMS	-	Dapat mengoperasikan <i>smartphone</i> Android
	Mengatur waktu periksa kesehatan lansia	<ul style="list-style-type: none"> • Hak akses halaman awal • Hak akses halaman atur jadwal 	
	Mengatur waktu minum obat lansia	<ul style="list-style-type: none"> • Hak akses halaman awal • Hak akses halaman atur obat 	
	Mengatur surasi konfirmasi keadaan	<ul style="list-style-type: none"> • Hak akses halaman awal • Hak akses halaman pengaturan 	

3.4 Spesifikasi Kebutuhan Perangkat Lunak

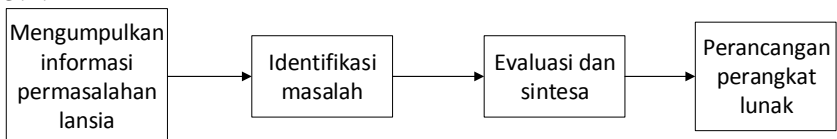
Kebutuhan aplikasi yang dibuat pada tugas akhir ini dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional.

3.4.1. Tahapan Spesifikasi Kebutuhan

Pada subab ini akan dijelaskan mengenai tahap yang digunakan dalam penentuan spesifikasi kebutuhan aplikasi. Terdapat empat langkah yang dilakukan sebagai berikut:

1. Mengumpulkan informasi masalah pada lansia
Pada tahap ini dilakukan studi literatur masalah-masalah lansia melalui *website* maupun buku.
2. Identifikasi masalah
Pada tahap ini dilakukan pemilihan masalah lansia yang akan diangkat dalam tugas akhir.
3. Evaluasi dan sintesa
Pada tahap ini dilakukan penentuan solusi dari permasalahan yang diangkat.
4. Perancangan perangkat lunak
Pada tahap ini dilakukan perancangan perangkat lunak dalam bentuk diagram-diagram maupun rancangan antarmuka.

Tahapan spesifikasi kebutuhan dapat digambarkan dalam Gambar 3.2.



Gambar 3.2 Tahapan Spesifikasi Kebutuhan

3.4.2. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fitur-fitur yang harus ada pada sistem dan reaksi dari sistem terhadap masukan yang diberikan. Daftar dari kebutuhan fungsional dari aplikasi yang dibuat pada tugas akhir ini dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kebutuhan Fungsional Aplikasi

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Mendaftarkan diri pada aplikasi	Pengasuh lansia dapat mendaftarkan diri pada aplikasi dengan memasukkan data-data yang diminta
F-002	Menambah data waktu minum obat	Pengasuh lansia dapat menambahkan jadwal waktu minum obat lansia
F-003	Mengubah data waktu minum obat	Pengasuh lansia dapat mengubah jadwal waktu minum obat lansia
F-004	Menghapus data waktu minum obat	Pengasuh lansia dapat menghapus jadwal waktu minum obat lansia
F-005	Menambah data waktu pemeriksaan kesehatan	Pengasuh lansia dapat menambahkan jadwal waktu pemeriksaan kesehatan lansia
F-006	Mengubah data waktu pemeriksaan kesehatan	Pengasuh lansia dapat mengubah jadwal waktu pemeriksaan kesehatan lansia
F-007	Menghapus data waktu pemeriksaan kesehatan	Pengasuh lansia dapat menghapus jadwal waktu pemeriksaan kesehatan lansia

F-008	Menyimpan nomor <i>handphone</i> pengasuh	Pengasuh lansia dapat menyimpan nomor <i>handphone</i> yang digunakan untuk menerima laporan dari aplikasi
F-009	Mengaktifkan fitur konfirmasi keadaan	Pengasuh lansia dapat mengaktifkan maupun menonaktifkan fitur konfirmasi keadaan dari aplikasi
F-010	Mengatur durasi konfirmasi keadaan	Pengasuh lansia dapat mengatur durasi konfirmasi keadaan
F-011	Memunculkan notifikasi pengingat	Lansia dapat menerima notifikasi pengingat waktu minum obat dan jadwal pemeriksaan kesehatan dari aplikasi
F-012	Memunculkan notifikasi konfirmasi	Lansia dapat menerima notifikasi konfirmasi keadaan dan konfirmasi terjatuh serta dapat memberikan jawabannya
F-013	Mengubah data profil	Pengasuh lansia dapat mengubah informasi profil yang dimasukkan saat <i>sign up</i>

3.4.3. Kebutuhan Non Fungsional

Aplikasi Lansia *Care* memiliki kebutuhan-kebutuhan non fungsional yang harus dipenuhi agar semua fitur aplikasi dapat berjalan dengan baik. Berikut adalah kebutuhan aplikasi Lansia *Care*:

1. Aplikasi harus di-*install* pada *smartphone* dengan *platform* Android.
2. Aplikasi harus di-*install* pada *smartphone* Android dengan sensor *accelerometer* dan *gyroscope* sudah terpasang sebagai *hardware*.
3. Aplikasi membutuhkan koneksi internet yang cepat agar fitur panggilan darurat dapat berjalan secara maksimal.
4. *Smartphone* dengan aplikasi *Lansia Care* tersimpan harus memiliki pulsa untuk mengirimkan pesan singkat maupun panggilan darurat.
5. *Text-to-speech* berbahasa Indonesia harus sudah *terinstall*.
6. Aplikasi menggunakan basis data SQLite sebagai penyimpanan datanya karena merupakan basis data yang ringan dan mudah digunakan.

3.3 Diagram Kasus Penggunaan

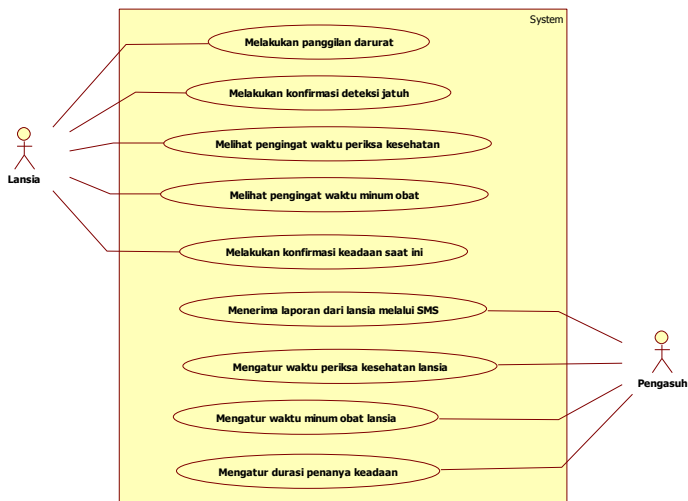
Adapun diagram kasus penggunaan yang telah dibuat dapat dilihat pada Gambar 3.3. Berdasarkan Gambar 3.3 aktor pada aplikasi adalah lansia dan pengasuhnya.

Lansia merupakan pengguna aplikasi, sedangkan pengasuh adalah aktor yang melakukan pengaturan pada aplikasi dan bertindak sebagai penerima informasi keadaan lansia dari sistem melalui pesan singkat (SMS). Adapun penjelasan aktivitas aktor sesuai Gambar 3.3 adalah sebagai berikut:

- a) Lansia merupakan aktor utama yang dapat melakukan aktivitas sebagai berikut:
 - Melakukan panggilan darurat.
 - Melakukan konfirmasi deteksi jatuh.
 - Melihat pengingat periksa kesehatan.
 - Melihat pengingat waktu minum obat.
 - Melakukan konfirmasi keadaan saat ini.

b) Pengasuh merupakan aktor yang diberikan tanggung jawab untuk melakukan pengaturan aplikasi dan penerima notifikasi, pengasuh dapat melakukan aktivitas sebagai berikut:

- Menerima laporan kondisi lansia melalui SMS.
- Mengatur waktu pemeriksaan kesehatan lansia.
- Mengatur waktu minum obat lansia.
- Mengatur durasi konfirmasi keadaan.



Gambar 3.3 Diagram Kasus Penggunaan

3.4 Diagram Alir Sistem

Pada bagian ini akan dijelaskan mengenai diagram alir dari metode-metode yang digunakan. Terdiri atas diagram alir teknik *windows sampling* dan *overlapping*, diagram alir teknik deteksi jatuh, serta diagram alir proses panggilan darurat. Data yang dihasilkan oleh sensor *accelerometer* dan *gyroscope* akan diolah dengan menggunakan teknik *windows sampling* dan *overlapping*. Kemudian data olahan tersebut akan dikenai proses pendeteksian

jatuh dengan menggunakan klasifikasi *k-Nearest Neighbour*. Panggilan darurat dilakukan dengan menggunakan *speech recognition*. Penjelasan mengenai ketiga diagram alir dijelaskan sebagai berikut:

3.4.1. Diagram Alir Teknik *Windows Sampling* dan *Overlapping*

Teknik *windows sampling* adalah suatu metode ekstraksi data yang digunakan untuk melakukan interpretasi data khusus. Metode ini akan mengekstraksi data dengan melakukan *sampling*, dimana setiap *windows* terdiri atas sekumpulan data yang merepresentasikan sebuah data. Ukuran *windows* dapat ditentukan sesuai kebutuhan. Pada tugas akhir ini digunakan lebar *window* sebesar 20.

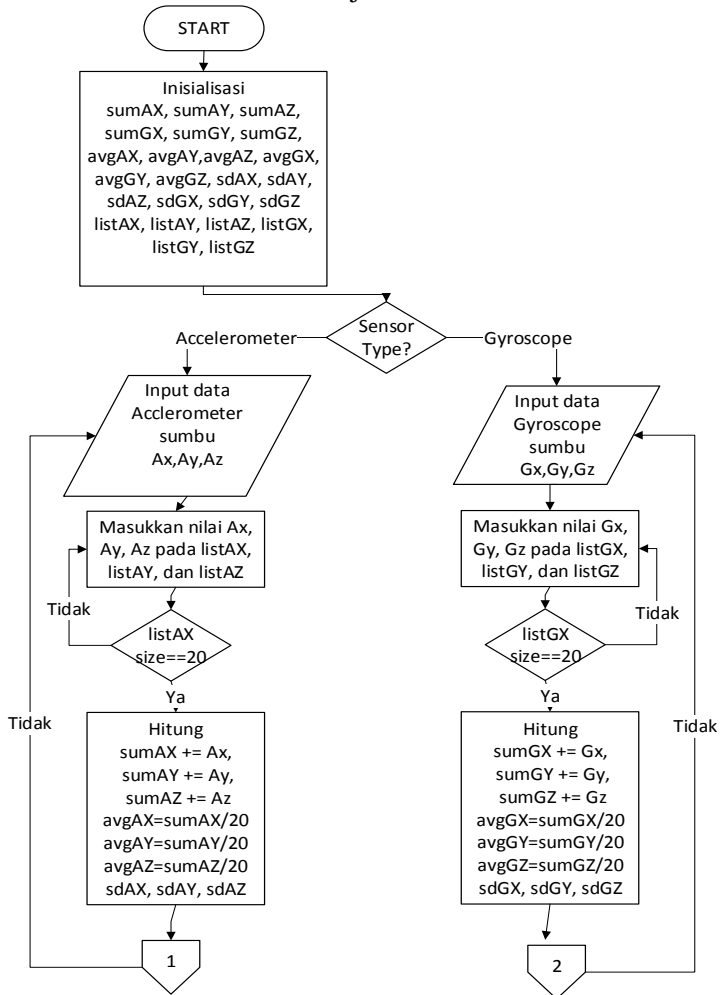
Teknik *overlapping* digunakan untuk menjaga konsistensi data. Teknik ini memperhitungkan setengah data baru dengan setengah data sebelumnya yang terdapat pada *window*. Alur kerja teknik *windows sampling* dan *overlapping* dapat digambarkan dalam diagram alir pada Gambar 3.4.

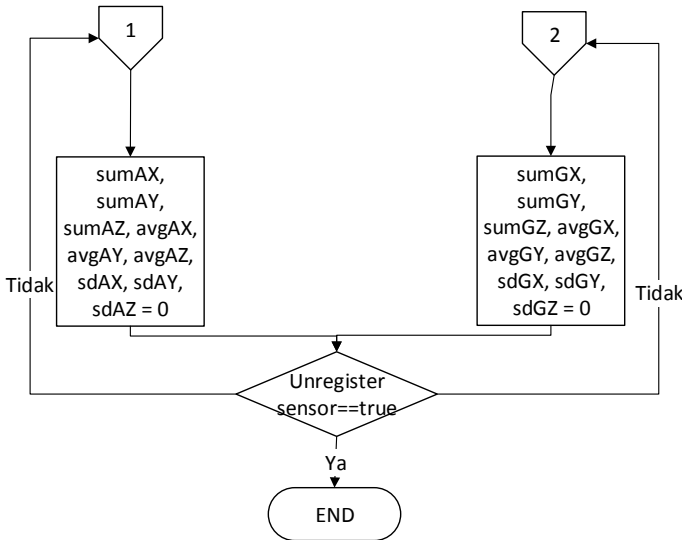
Berikut adalah rincian dari tiap-tiap tahapan dalam diagram alir pada Gambar 3.4:

1. Inisialisasi variabel-variabel yaitu *sum*, *average*, dan *standard deviation*, serta *list* penyimpanan hasil pembacaan sensor untuk tiga sumbu yaitu *x*, *y*, dan *z* untuk kedua sensor yaitu sensor *accelerometer* dan *gyroscope*.
2. Dilakukan pemeriksaan apakah sensor yang terbaca adalah sensor *accelerometer* atau *gyroscope*.
3. Untuk tiap sensor, masukkan hasil perubahan sensor ke dalam *list* penyimpanan untuk masing-masing sumbunya.
4. Dilakukan pemeriksaan apakah *list* penyimpanan telah mencapai ukuran *window*, yaitu 20. Apabila tidak, akan tetap dilakukan penyimpanan hasil perubahan sensor pada *list*. Apabila ya, maka akan menuju tahapan selanjutnya.
5. Ketika *list* penyimpan telah mencapai ukuran 20, akan dilakukan penghitungan rata-rata dengan menggunakan

Persamaan 2.1 dan simpangan baku dengan menggunakan Persamaan 2.2.

6. Dilakukan teknik *overlapping*, yaitu menghapus setengah ukuran *window* pada *list*.
7. Menginisialisasi kembali variabel *sum*, *average*, dan *standard deviation* menjadi 0 kembali.





Gambar 3.4 Diagram Alir Teknik *Windows Sampling* dan *Overlapping*

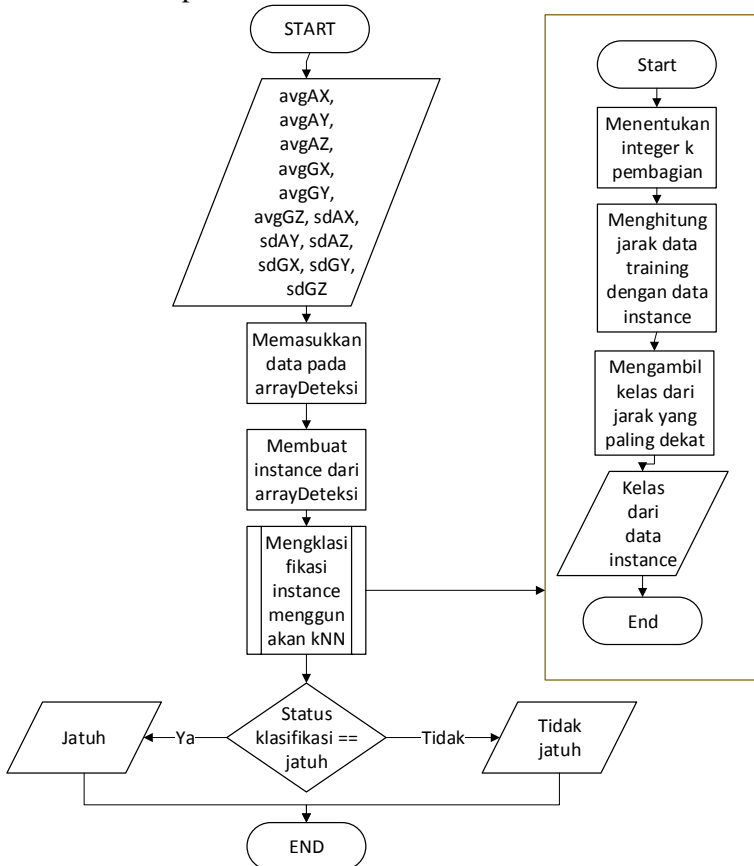
3.4.1. Diagram Alir Teknik Deteksi Jatuh

Pendeteksian jatuh dihasilkan melalui analisa data dari sensor *accelerometer* dan *gyroscope*. Sensor *accelerometer* akan mendeteksi percepatan terhadap sumbu x , y , dan z . Sedangkan sensor *gyroscope* mendeteksi perputaran atau orientasi terhadap sumbu x , y , dan z . Hasil pengambilan data *training* sensor akan dilabeli terjatuh atau tidak, kemudian data akan dikenai klasifikasi *k-Nearest Neighbour*. Alur kerja deteksi jatuh dapat digambarkan dalam Gambar 3.5. Teknik deteksi jatuh dilakukan setelah *window* telah terisi penuh.

Berikut adalah rincian dari tiap-tiap tahapan dalam diagram alir pada Gambar 3.5:

1. Mengambil data rata-rata dan simpangan baku dari penghitungan pada teknik *windows sampling* dan *overlapping*.

2. Menginisialisasi *array double* dan memasukkan variabel rata-rata dan simpangan baku ke dalam *array*.
3. Membuat *instance* dari *array double* pada tahapan sebelumnya.
4. Melakukan klasifikasi *instance* dengan menggunakan *classifier k-Nearest Neighbour* yaitu dengan mencari kelas dari titik yang paling dekat dengan titik *instance*.
Mendapatkan kelas klasifikasi dari *instance*.



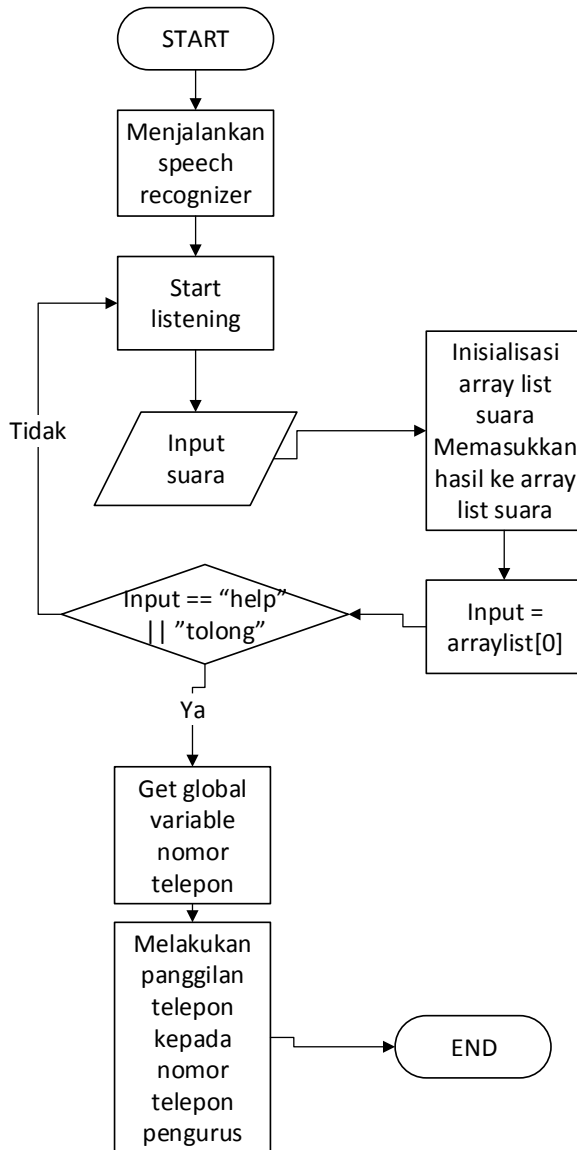
Gambar 3.5 Diagram Alir Teknik Deteksi Jatuh

3.4.2. Diagram Alir Proses Panggilan Darurat

Panggilan darurat pada aplikasi dilakukan dengan menggunakan *speech recognition* yang berjalan pada *service*. *Speech recognition* akan terus berjalan dan mendeteksi suara. Pengguna yang akan melakukan panggilan darurat dapat mengucapkan kata kunci yaitu “*help*” ataupun “tolong” dan aplikasi pun akan melakukan panggilan telepon pada pengasuh lansia. Alur kerja proses panggilan darurat dapat digambarkan dalam diagram alir Gambar 3.6.

Berikut adalah rincian dari tiap-tiap tahapan dalam diagram alir pada Gambar 3.6:

1. Menjalankan *speech recognizer service*.
2. *Speech recognizer* akan mulai melakukan penangkapan suara.
3. Ketika suara tertangkap maka akan dilakukan inisialisasi *list* untuk penyimpanan suara yang tertangkap, kemudian suara yang ditangkap akan dimasukkan ke dalam *list* tersebut.
4. Mengambil indeks pertama pada *list* untuk dilakukan pengecekan apakah suara yang paling jelas ditangkap adalah kata kunci panggilan darurat yaitu “*help*” atau “tolong”.
5. Apabila merupakan kata kunci panggilan darurat, akan diambil variabel nomor telepon yang telah dimasukkan.
6. Melakukan panggilan telepon pada nomor telepon yang telah didapat.



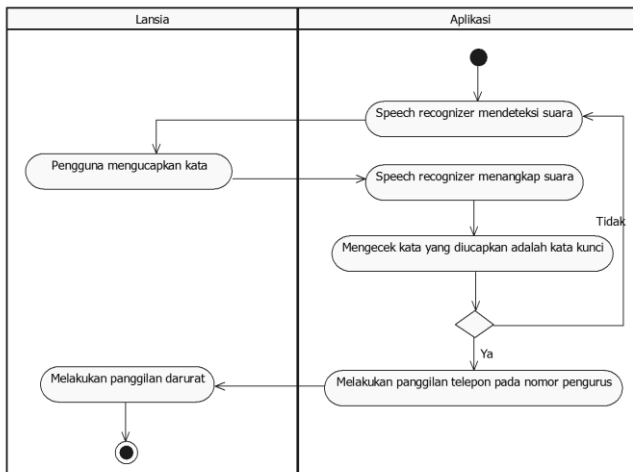
Gambar 3.6 Diagram Alir Proses Panggilan Darurat

3.5 Diagram Aktivitas Sistem

Diagram aktivitas adalah diagram yang menggambarkan garis besar alur dari tiap-tiap *use case* yang telah dijelaskan pada Gambar 3.3. Diagram aktivitas menggambarkan proses atau aktivitas dari level atas secara umum. Pada subbab ini akan dijelaskan mengenai diagram aktivitas yang terdapat pada Tugas Akhir.

3.5.1. Diagram Aktivitas Melakukan Panggilan Darurat

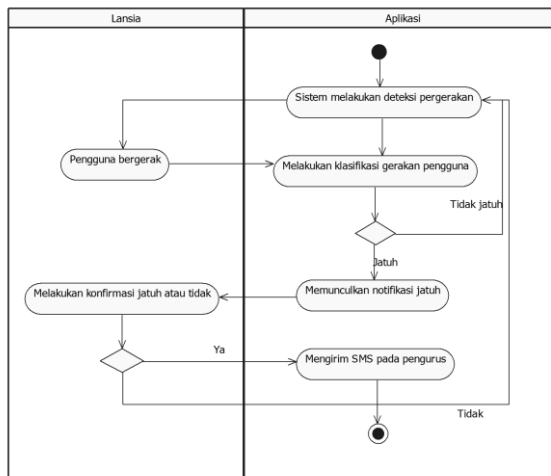
Diagram aktivitas melakukan panggilan darurat menjelaskan tentang bagaimana lansia dapat melakukan panggilan darurat secara otomatis dengan mengucapkan kata kunci yaitu “*help*” ataupun “*tolong*”. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.7.



Gambar 3.7 Diagram Aktivitas Melakukan Panggilan Darurat

3.5.2. Diagram Aktivitas Melakukan Konfirmasi Deteksi Jatuh

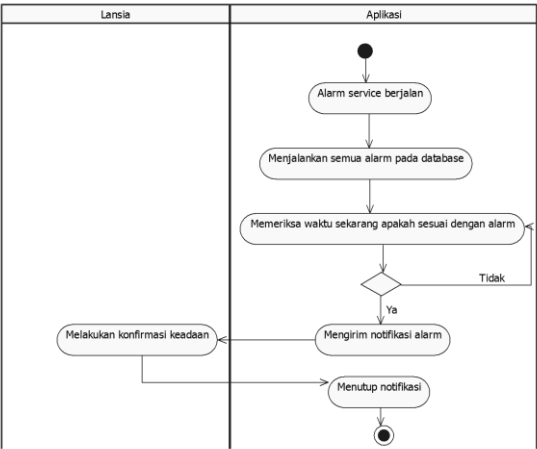
Diagram aktivitas melakukan konfirmasi deteksi jatuh menjelaskan bagaimana lansia melakukan konfirmasi apabila pergerakan pengguna terdeteksi terjatuh. Hal ini dimaksudkan untuk menghindari *false positive* dari sistem. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.8.



Gambar 3.8 Diagram Aktivitas Melakukan Konfirmasi Deteksi Jatuh

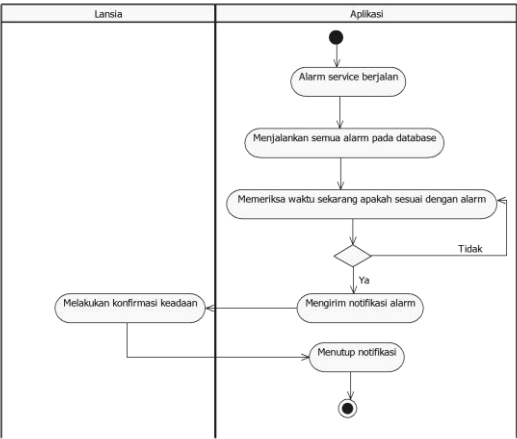
3.5.3. Diagram Aktivitas Melihat Pengingat Waktu Periksa Kesehatan

Gambar 3.9 merupakan diagram aktivitas melihat pengingat waktu periksa kesehatan yang menjelaskan bagaimana lansia mendapatkan notifikasi waktu periksa kesehatan yang telah diatur oleh pengasuhnya. Lansia hanya perlu menutup tampilan notifikasi yang muncul dengan menekan tombol *tutup_notifikasi_pengingat* yang ada di bagian bawah layar.



Gambar 3.9 Diagram Aktivitas Melihat Pengingat Waktu Periksa Kesehatan

3.5.4. Diagram Aktivitas Melihat Pengingat Waktu Minum Obat

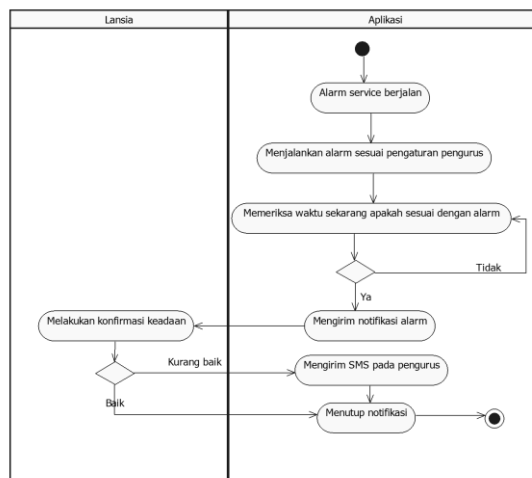


Gambar 3.10 Aktivitas Melihat Pengingat Waktu Minum Obat

Gambar 3.10 merupakan diagram aktivitas melihat pengingat waktu minum obat yang menjelaskan bagaimana lansia mendapatkan notifikasi waktu minum obat yang telah diatur oleh pengasuhnya.

3.5.5. Diagram Aktivitas Melakukan Konfirmasi Keadaan Saat Ini

Diagram aktivitas melakukan konfirmasi keadaan saat ini menjelaskan bagaimana lansia melakukan konfirmasi keadaannya pada waktu yang telah ditentukan aplikasi dan dengan durasi konfirmasi yang telah ditentukan pengasuhnya. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.11.

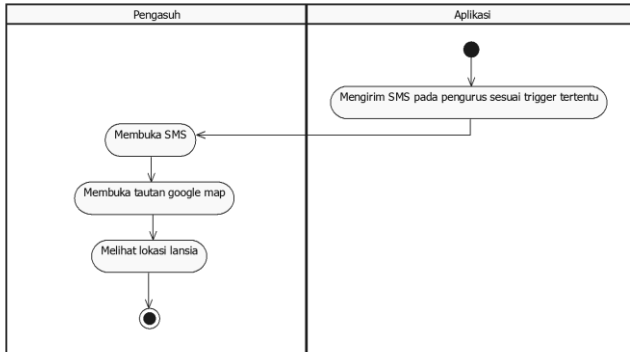


Gambar 3.11 Diagram Aktivitas Melakukan Konfirmasi Keadaan Saat Ini

3.5.6. Diagram Aktivitas Menerima Laporan Lansia Melalui SMS

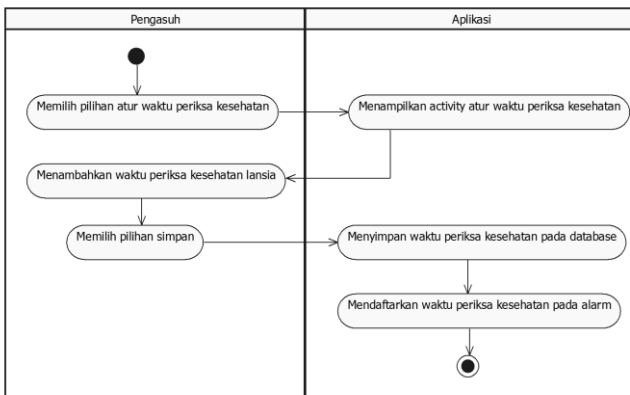
Diagram aktivitas menerima laporan lansia melalui SMS menjelaskan bagaimana pengasuh lansia menerima laporan lansia

baik laporan keadaan maupun laporan lansia terjatuh dalam bentuk pesan singkat (SMS) yang berisi pesan maupun tautan Google Maps yang menunjukkan posisi lansia saat ini. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.12.



Gambar 3.12 Diagram Aktivitas Menerima Laporan Lansia Melalui SMS

3.5.7. Diagram Aktivitas Mengatur Waktu Periksa Kesehatan Lansia

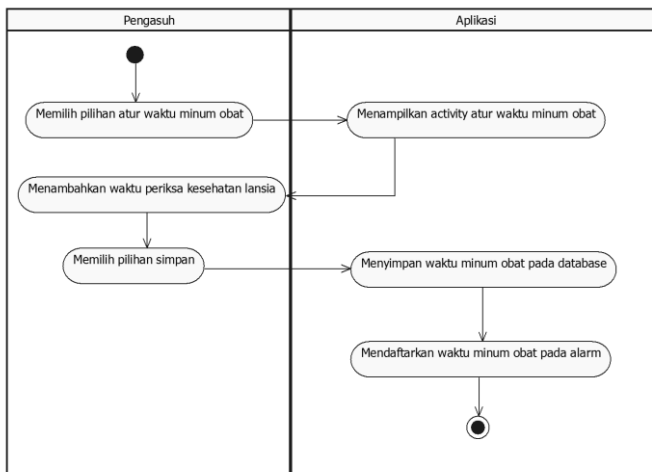


Gambar 3.13 Diagram Aktivitas Mengatur Waktu Periksa Kesehatan Lansia

Diagram aktivitas mengatur waktu pemeriksaan kesehatan lansia menjelaskan bagaimana pengasuh melakukan pengaturan pada aplikasi terkait dengan waktu pemeriksaan kesehatan lansia. Aplikasi akan mengingatkan lansia melalui notifikasi pada tanggal yang telah diatur oleh pengasuh setiap bulannya. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.13.

3.5.8. Diagram Aktivitas Mengatur Waktu Minum Obat Lansia

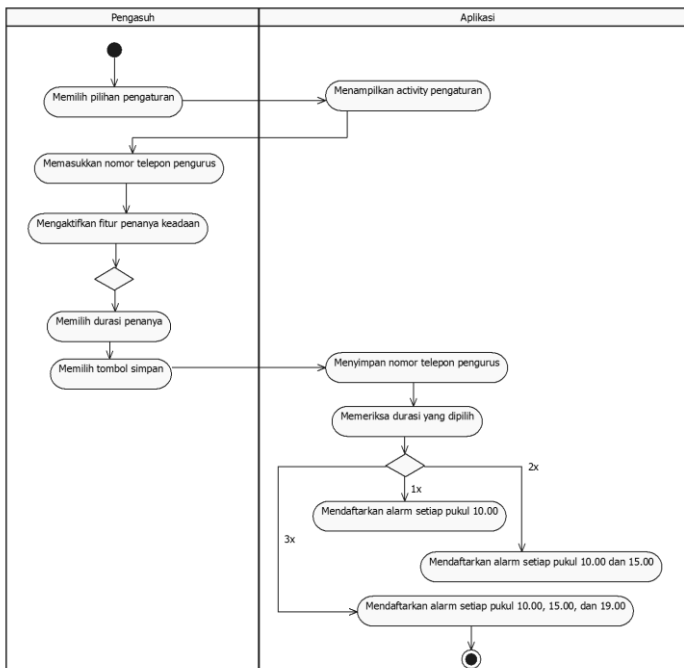
Diagram aktivitas mengatur waktu minum obat lansia menjelaskan bagaimana pengasuh melakukan pengaturan pada aplikasi terkait dengan waktu minum obat lansia. Aplikasi akan mengingatkan lansia melalui notifikasi pada waktu yang telah diatur oleh pengasuh setiap hari. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.14.



Gambar 3.14 Mengatur Waktu Minum Obat Lansia

3.5.9. Diagram Aktivitas Mengatur Durasi Konfirmasi Keadaan

Diagram aktivitas mengatur durasi konfirmasi keadaan menjelaskan bagaimana pengasuh melakukan pengaturan durasi untuk konfirmasi keadaan. Terdapat 3 pilihan durasi pada aplikasi, yaitu 1 kali, 2 kali, dan 3 kali. Rancangan diagram aktivitasnya dapat dilihat pada Gambar 3.15.



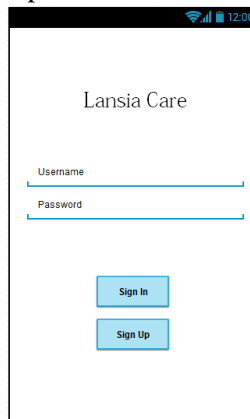
Gambar 3.15 Diagram Aktivitas Mengatur Durasi Konfirmasi Keadaan

3.6 Rancangan Antarmuka

Rancangan antarmuka pada aplikasi ini ditujukan untuk pengguna *smartphone* dengan sistem operasi Android.

3.6.1. Rancangan Antarmuka Halaman *Log In*

Gambar 3.16 merupakan rancangan antarmuka halaman yang muncul pertama kali saat pengguna membuka aplikasi. *Log in* digunakan agar orang lain tidak dapat melakukan pengaturan pada aplikasi kecuali pengasuh lansia. Apabila pengasuh belum memiliki akun, maka dapat memilih menu *sign up*.

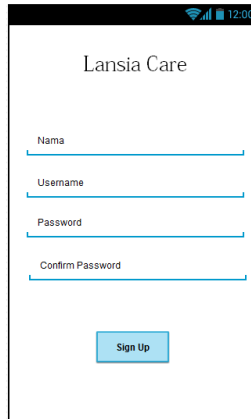
The image shows a mobile application login screen titled "Lansia Care". At the top, there is a status bar with signal, Wi-Fi, and battery icons, and the time "12:00". Below the title, there are two input fields: "Username" and "Password", each with a blue underline. At the bottom, there are two blue buttons: "Sign In" and "Sign Up".

Gambar 3.16 Rancangan Antarmuka Halaman *Sign In*

Pengguna diminta untuk memasukkan *username* dan *password* sesuai dengan yang telah didaftarkan.

3.6.2. Rancangan Antarmuka Halaman *Sign Up*

Gambar 3.17 merupakan rancangan antarmuka halaman yang muncul saat pengguna memilih pilihan *sign up* pada halaman *log in*. Halaman ini digunakan untuk mendaftarkan akun pengguna pada aplikasi.



Gambar 3.17 Rancangan Antarmuka Halaman *Sign Up*

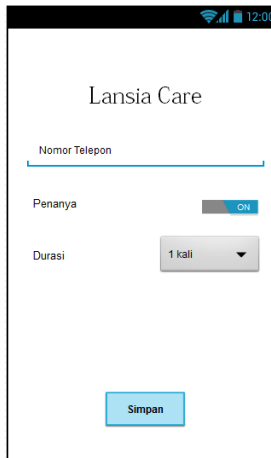
3.6.3. Rancangan Antarmuka Halaman Utama



Gambar 3.18 Rancangan Antarmuka Halaman Utama

Gambar 3.18 merupakan rancangan antarmuka halaman yang muncul saat pengguna berhasil *log in* dalam aplikasi. Terdapat empat pilihan yang dapat dipilih yaitu pengaturan, jadwal obat, jadwal pemeriksaan, dan ubah profil.

3.6.4. Rancangan Antarmuka Halaman Pengaturan

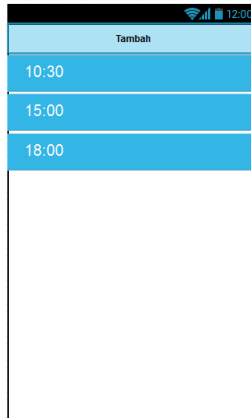


Gambar 3.19 Rancangan Antarmuka Halaman Pengaturan

Gambar 3.19 merupakan rancangan antarmuka yang muncul saat pengguna memilih pilihan pengaturan pada halaman utama. Pengasuh dapat memasukkan nomor telepon yang nantinya digunakan sebagai penerima pesan singkat. Selain itu, pengasuh dapat mengaktifkan fitur konfirmasi keadaan dan durasinya.

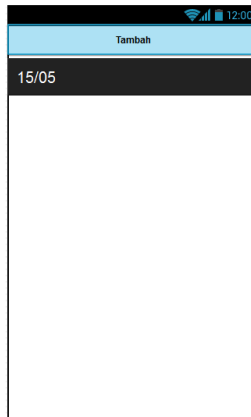
3.6.5. Rancangan Antarmuka Halaman Atur Obat

Gambar 3.20 merupakan rancangan antarmuka yang muncul saat pengguna memilih pilihan jadwal obat pada halaman utama. Halaman ini menampilkan daftar waktu minum obat yang telah diatur oleh pengasuh lansia. Pengasuh dapat menambahkan jadwal waktu minum obat dengan memilih pilihan tambah.



Gambar 3.20 Rancangan Antarmuka Halaman Atur Obat

3.6.6. Rancangan Antarmuka Halaman Atur Jadwal



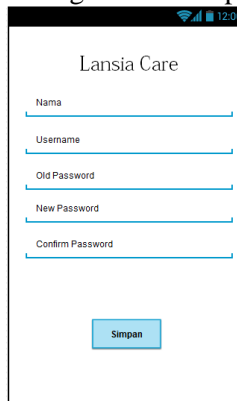
Gambar 3.21 Rancangan Antarmuka Halaman Atur Jadwal

Gambar 3.21 merupakan rancangan antarmuka yang muncul saat pengguna memilih pilihan jadwal pemeriksaan pada halaman utama. Halaman ini menampilkan daftar jadwal pemeriksaan kesehatan lansia yang telah diatur oleh pengasuh

lansia. Pengasuh dapat menambahkan jadwal pemeriksaan kesehatan dengan memilih pilihan tambah.

3.6.7. Rancangan Antarmuka Halaman Ubah Profil

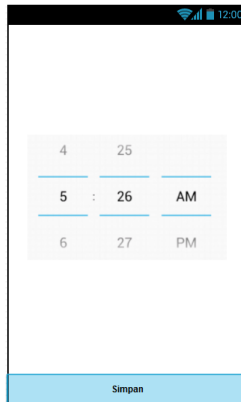
Gambar 3.22 merupakan rancangan antarmuka yang muncul setelah pengasuh memilih pilihan ubah profil pada halaman utama. Pengasuh dapat memperbarui datanya dan menyimpannya kembali dengan memilih pilihan simpan.

The image shows a mobile application interface for 'Lansia Care'. At the top, there's a status bar with signal, Wi-Fi, and battery icons, and the time 12:00. Below the status bar, the app title 'Lansia Care' is centered. Underneath the title, there are five input fields, each with a label above it: 'Nama', 'Username', 'Old Password', 'New Password', and 'Confirm Password'. Each input field has a blue underline. At the bottom of the form, there is a blue button with the text 'Simpan' (Save) in white.

Gambar 3.22 Rancangan Antarmuka Halaman Ubah Profil

3.6.8. Rancangan Antarmuka Halaman Tambah Waktu Minum Obat

Gambar 3.23 merupakan rancangan antarmuka yang muncul setelah pengguna memilih pilihan tambah pada halaman atur obat. Pengasuh dapat mengatur waktu minum obat dengan mengubah waktu pada *timepicker* dan menyimpannya dengan memilih pilihan simpan.

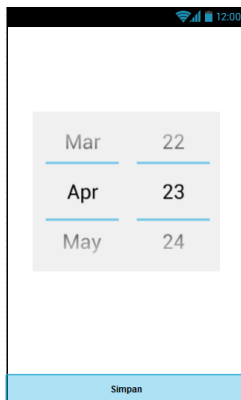


4	25	
5	:	26 AM
6	27	PM

Simpan

Gambar 3.23 Rancangan Antarmuka Halaman Tambah Waktu Minum Obat

3.6.9. Rancangan Antarmuka Halaman Tambah Jadwal Pemeriksaan Kesehatan Lansia



Mar	22
Apr	23
May	24

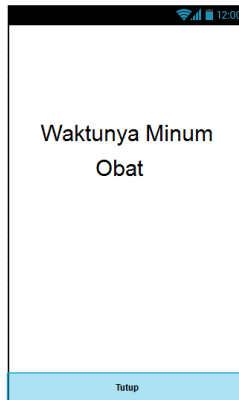
Simpan

Gambar 3.24 Rancangan Antarmuka Halaman Tambah Jadwal Pemeriksaan Kesehatan Lansia

Gambar 3.24 merupakan rancangan antarmuka yang muncul setelah pengguna memilih pilihan tambah pada halaman atur jadwal. Pengasuh dapat mengatur jadwal pemeriksaan

kesehatan dengan mengubah tanggal pada *datepicker* dan menyimpannya dengan memilih pilihan simpan.

3.6.10. Rancangan Antarmuka Notifikasi Waktu Minum Obat



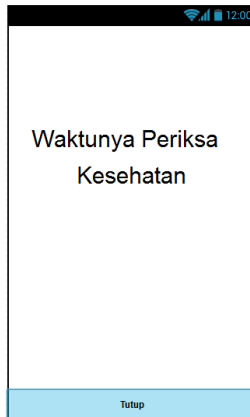
Gambar 3.25 Rancangan Antarmuka Notifikasi Waktu Minum Obat

Gambar 3.25 merupakan rancangan antarmuka yang muncul setiap waktu saat ini menunjukkan waktu yang sama dengan waktu minum obat yang telah diatur oleh pengasuh. Notifikasi yang muncul menggunakan *text-to-speech* untuk mengingatkan lansia bahwa saat ini adalah waktu minum obat. Lansia dapat mematikan notifikasi ini dengan memilih pilihan tutup.

3.6.11. Rancangan Antarmuka Notifikasi Jadwal Pemeriksaan Kesehatan

Gambar 3.26 merupakan rancangan antarmuka yang muncul setiap tanggal saat ini menunjukkan tanggal yang sama dengan jadwal pemeriksaan kesehatan yang telah diatur oleh pengasuh. Notifikasi yang muncul menggunakan *text-to-speech* untuk mengingatkan lansia bahwa saat ini adalah saat untuk periksa

kesehatan. Lansia dapat mematikan notifikasi ini dengan memilih pilihan tutup.



Gambar 3.26 Rancangan Antarmuka Notifikasi Jadwal Pemeriksaan Kesehatan

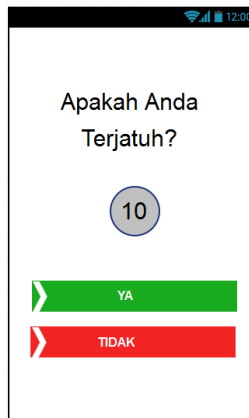
3.6.12. Rancangan Antarmuka Notifikasi Konfirmasi Keadaan



Gambar 3.27 Rancangan Antarmuka Notifikasi Konfirmasi Keadaan

Gambar 3.27 merupakan rancangan antarmuka yang muncul pada waktu-waktu tertentu sesuai dengan durasi yang dipilih oleh pengasuh sebelumnya. Lansia dapat menjawab dengan menggeser tombol baik apabila lansia merasa kondisi kesehatannya baik, sebaliknya menggeser tombol kurang baik apabila lansia merasa kondisi kesehatannya kurang baik.

3.6.13. Rancangan Antarmuka Notifikasi Konfirmasi Jatuh



Gambar 3.28 Rancangan Antarmuka Notifikasi Konfirmasi Jatuh

Gambar 3.28 merupakan rancangan antarmuka yang muncul saat lansia terdeteksi terjatuh. Notifikasi ini dimaksudkan untuk menghindari *false positive* dari proses deteksi jatuh. Lansia dapat memilih pilihan tidak apabila lansia tidak terjatuh, selain itu juga terdapat hitung mundur, apabila lansia tidak melakukan konfirmasi selama hitung mundur maka lansia dianggap terjatuh dan aplikasi akan mengirimkan notifikasi terjatuh pada pengasuh melalui pesan singkat.

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Aplikasi dibuat pada lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1. Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan aplikasi adalah sebagai berikut:

- Laptop
 - *Processor* Intel® Core™ i7-4500U CPU @ 1.80GHz 2.39 GHz
 - RAM 8.00 GB
- Perangkat *mobile* Samsung Galaxy Note II GT-N7100
 - Sistem Operasi Android versi 4.4.2
 - *Processor* Quad Core 1.6GHz
 - *Memory Card Slot* microSD up to 64GB
 - *Internal memory* 16GB
 - 2 GB RAM
 - Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi Direct, DLNA, hotspot

4.1.2. Lingkungan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada lingkungan perancangan dan pengembangan aplikasi adalah sebagai berikut:

- Sistem Operasi Windows 8.1 64 bit
- Android Studio Development Tools 1.0.1
- Star UML versi 5.0.2.1570
- Pencil versi 2.0.3
- Microsoft Office Visio 2013
- Microsoft Office Word 2013

4.2 Implementasi Proses

Pada subbab ini akan dijelaskan tentang proses-proses yang dilakukan oleh aplikasi. Proses-proses yang dilakukan aplikasi mulai dari inialisasi *sensor listener* hingga pengiriman notifikasi SMS kepada pengasuh.

4.2.1. Implementasi Inialisasi *SensorListener*

```

1:  initialize SENSOR MANAGER
2:  initialize sensor TYPE_ACCELEROMETER
3:  initialize sensor TYPE_GYROSCOPE
4:  set SENSOR_SERVICE on
5:  register      sensor      TYPE_ACCELEROMETER
   listener
6:  register sensor TYPE_GYROSCOPE listener

```

Gambar 4.1 Pseudocode Implementasi Inialisasi *SensorListener*

Gambar 4.1 merupakan *pseudocode* implementasi inialisasi *SensorEventListener* untuk sensor *accelerometer* dan *gyroscope*. Hal pertama yang dilakukan adalah melakukan inialisasi *Sensor Manager*. Inialisasi *Sensor Manager* digunakan untuk instansiasi layanan sensor, dimana hal tersebut berfungsi untuk membuka *port listener* sensor. Setelah *port*

listener terbuka, maka *Sensor Manager* dapat melakukan registrasi *Sensor Listener*.

4.2.2. Implementasi Mendapatkan Data Sensor Accelerometer dan Gyroscope

Gambar 4.2 merupakan *pseudocode* implementasi untuk mendapatkan data sensor *accelerometer* dan *gyroscope*. Fungsi *onSensorChanged()* berfungsi untuk mendapatkan nilai sensor ketika nilai sensor berubah.

```

onSensorChanged()
1:  if sensor type is accelerometer then
2:      get value acc_x
3:      get value acc_y
4:      get value acc_z
5:  end if
6:  if sensor type is gyroscope then
7:      get value gyro_x
8:      get value gyro_y
9:      get value gyro_z
10: end if

```

Gambar 4.2 Pseudocode Implementasi Mendapatkan Data Sensor Accelerometer dan Gyroscope

Dua buah sensor yang telah diinisialisasi akan dideteksi perubahannya pada fungsi ini. Apabila sistem mendeteksi perubahan pada sensor bertipe *accelerometer* maka nilai sensor pada sumbu *x*, *y*, dan *z* akan diberikan pada variabel *acc_x*, *acc_y*, dan *acc_z*. Apabila sistem mendeteksi perubahan pada sensor bertipe *gyroscope* maka nilai sensor pada sumbu *x*, *y*, dan *z* akan diberikan pada variabel *gyro_x*, *gyro_y*, dan *gyro_z* kemudian hasil dari tiap sumbu masing-masing sensor dimasukkan ke dalam *list*.

4.2.3. Implementasi Teknik *Windows Sampling* dan *Overlapping*

```

onSensorChanged()
1:  if list size < 20
2:      add sensorX to listSumbuX
3:      add sensorY to listSumbuY
4:      add sensorZ to listSumbuZ
5:  end if
6:  else
7:      hitungAverage()
8:      hitungStDev()
9:      for i=0 to list size
10:         remove element listSumbuX at i
11:         remove element listSumbuY at i
12:         remove element listSumbuZ at i
13:      end if

```

Gambar 4.3 Pseudocode Implementasi Teknik *Windows Sampling* dan *Overlapping*

Gambar 4.3 merupakan *pseudocode* implementasi teknik *windows sampling* dan *overlapping*. Teknik ini digunakan untuk menghindari data yang divergen dan juga dapat memperhitungkan pembacaan data sebelumnya. Pertama, tiap-tiap sensor akan mendapatkan nilai yang disimpan dalam variabel tertentu bertipe data *double*. Selama lebar *window* masih cukup, maka nilai-nilai tersebut akan disimpan dalam *list* untuk tiap-tiap sumbunya. Sedangkan, apabila *window* telah penuh maka akan dilakukan penghitungan rata-rata dan simpangan baku, serta menghapus setengah data *window*.

4.2.4. Implementasi Menghitung Rataan Data Sensor

Gambar 4.4 merupakan *pseudocode* menghitung rata-rata data dalam satu *window*. Lebar *window* yang digunakan pada tugas akhir ini sebesar 20. Rata-rata digunakan untuk menghindari data sensor yang divergen.

```

Input: listSumbuX, listSumbuY, listSumbuZ
Output: avgX, avgY, avgZ
hitungAverage()
1: initialize sumX, sumY, sumZ to 0
2: initialize avgX, avgY, avgZ to 0
3: for i=0 to list size
4:     sumX += get listSumbuX at i
5:     sumY += get listSumbuY at i
6:     sumZ += get listSumbuZ at i
7: end for
8: avgX = sumX/20
9: avgY = sumY/20
10: avgZ = sumZ/20

```

Gambar 4.4 Pseudocode Implementasi Menghitung Rataan Data Sensor

4.2.5. Implementasi Menghitung Simpangan Baku Data Sensor

```

Input: listSumbuX, listSumbuY, listSumbuZ,
avgX, avgY, avgZ
Output: stDevX, stDevY, stdevZ
hitungStDev()
1: initialize stDevX, stDevY, stDevZ to 0
2: for i=0 to list size
3:     stDevX = sqrt(pow(listSumbuX at i -
                        avgX),2)/20-1)
4:     stDevY = sqrt(pow(listSumbuY at i -
                        avgY),2)/20-1)
5:     stDevZ = sqrt(pow(listSumbuZ at i -
                        avgZ),2)/20-1)
6: end for

```

Gambar 4.5 Pseudocode Implementasi Menghitung Simpangan Baku Data Sensor

Gambar 4.5 merupakan *pseudocode* menghitung simpangan baku data sensor untuk menambah akurasi dari

klasifikasi data. Simpangan baku dihitung dengan menggunakan Persamaan 2.2.

4.2.6. Implementasi Membuat *Classifier*

```
1: initialize reader
2: initialize dataset trainingData
3: initialize classifier kNearestNeighbour
4: open asset data training
5: read asset into reader
6: load reader to dataset
7: build classifier kNearestNeighbour using
   dataset
```

Gambar 4.6 Pseudocode Implementasi Membuat *Classifier*

Gambar 4.6 merupakan *pseudocode* implementasi membuat *classifier*. *Classifier* yang digunakan pada tugas akhir adalah *k-Nearest Neighbour*. Pertama, *data training* dalam format *.csv* akan dibaca oleh *file reader*. Kemudian data yang telah dibaca dipindahkan ke dalam tipe data *dataset*. Dari *dataset* tersebut akan dibangun *classifier k-Nearest Neighbour*.

4.2.7. Implementasi Klasifikasi Data Sensor

Gambar 4.7 merupakan *pseudocode* implementasi mengklasifikasi data sensor menggunakan *classifier k-Nearest Neighbour*. Untuk setiap *window*, data akan diklasifikasi. Hasil pengklasifikasian data akan disimpan dalam bentuk *string* sesuai dengan hasil kelas klasifikasi. Jika kelas klasifikasi menunjukkan jatuh maka aplikasi akan memanggil *activity* konfirmasi jatuh dan menghentikan sementara proses klasifikasi hingga pengguna memberi konfirmasi atau apabila tidak ada konfirmasi, klasifikasi akan berjalan kembali setelah 10 detik. Sebaliknya apabila kelas klasifikasi menunjukkan tidak jatuh maka proses klasifikasi akan berjalan terus.


```

classifyData()
1: initialize array double
2: initialize instance
3: initialize string status
4: store avgX, avgY, avgZ, stDevX, stDevY,
   stDevZ to array double
5: create dense instance from array double
6: classify instance using kNearestNeighbour
7: store classification result to status
8: if status equals jatuh
9:     initialize intent
10:    start intent activity
11:    stop classification
12: end if

```

Gambar 4.7 Pseudocode Implementasi Klasifikasi Data Sensor

4.2.8. Implementasi Mendapatkan Lokasi Pengguna

Gambar 4.8 merupakan *pseudocode* mendapatkan lokasi pengguna saat ini. Lokasi pengguna didapatkan dengan pembacaan GPS (*Global Positioning System*). Pembacaan lokasi pengguna saat ini berjalan pada *background service*. Adapun langkah-langkah dalam pembacaan lokasi pengguna adalah sebagai berikut:

1. Pembacaan GPS diawali dengan inisialisasi *LocationManager* dan *LocationListener*. *Location* diinisialisasi *null*, hal ini menandakan bahwa sebelum aplikasi dimulai maka tidak ada nilai pada *location*. *Location* berisi nilai *latitude* dan *longitude*.
2. Selama aplikasi mulai melakukan pendeteksian, dilakukan pengecekan *location* dan *provider*. Pada tugas akhir ini *provider* yang digunakan adalah GPS *provider* dan *network provider*. Pengecekan yang dilakukan mengenai perubahan *location* yang meliputi perubahan nilai *latitude* dan *longitude* serta ketersediaan *provider*. Jika sistem mendeteksi perubahan nilai *latitude* dan *longitude* serta status *provider* dalam kondisi aktif, maka beralih pada langkah selanjutnya.

- Langkah terakhir yaitu pembaruan nilai *location*. Nilai *location* diperbarui sesuai dengan posisi lansia saat itu. Selain itu, nilai *location* diperbarui setiap sepuluh meter dan waktu satu menit.

```

getLocation()
1:  initialize location
2:  set location to null
3:  initialize location manager
4:  initialize location listener
5:  while sensor detection on
6:      location manager get gps provider
7:      location manager get network provider
8:      if gps provider enabled
9:          request location update
10:         update value of location
11:     end if
12:     if network provider enabled
13:         request location update
14:         update value of location
15:     end if
16: end while

```

Gambar 4.8 Pseudocode Implementasi Mendapatkan Lokasi Pengguna

4.2.9. Implementasi Mendaftarkan Pengingat Minum Obat

Gambar 4.9 merupakan *pseudocode* mendaftarkan pengingat waktu minum obat pada *alarm*. Adapun langkah-langkah dalam mendaftarkan waktu minum obat pada *alarm* adalah sebagai berikut:

- Pertama adalah dengan menginisialisasi *timepicker* dan *calendar*. Setelah pengguna memilih waktu yang diinginkan pada *timepicker* maka akan dilakukan pengaturan jam dan menit pada *calendar* sesuai dengan masukan pengguna.
- Inisialisasi *pendingIntent* dan *AlarmManager*. *PendingIntent* digunakan untuk mendaftarkan *alarm* pada *broadcast receiver*. Sedangkan *AlarmManager* digunakan untuk mengatur kapan *alarm* akan diluncurkan.

3. Mengatur interval *alarm* yang akan diluncurkan setiap harinya pada waktu yang telah diatur pengasuh.
4. Menyimpan data *alarm* pada *database*.

```
tambahAlarm()
1: initialize timePicker
2: initialize calendar
3: get hour from timePicker
4: get minute from timePicker
5: set calendar hour
6: set calendar minute
7: initialize PendingIntent
8: initialize AlarmManager
9: call getInterval() function
10: set repeating alarm with interval
```

Gambar 4.9 Pseudocode Implementasi Mendaftarkan Pengingat Minum Obat

```
getInterval()
1: initialize int day, hour, minute, second,
   and millisecond
2: initialize int interval
3: set day to 1
4: set hour to 24
5: set minute to 60
6: set second to 60
7: set millisecond to 1000
8: interval = day * hour * minute * second *
   millisecond
9: return interval
```

Gambar 4.10 Pseudocode Mendapatkan Interval Alarm Waktu Minum Obat

4.2.10. Implementasi Mendaftarkan Pengingat Pemeriksaan Kesehatan

Gambar 4.11 merupakan *pseudocode* mendaftar pengingat waktu pemeriksaan kesehatan pada *alarm*. Adapun langkah-langkah dalam mendaftar waktu pemeriksaan kesehatan pada *alarm* adalah sebagai berikut:

1. Pertama adalah dengan menginisialisasi *datepicker* dan *calendar*. Setelah pengguna memilih tanggal yang diinginkan pada *datepicker* maka akan dilakukan pengaturan tanggal dan bulan pada *calendar* sesuai dengan masukan pengguna.
2. Inisialisasi *pendingIntent* dan *AlarmManager*. *PendingIntent* digunakan untuk mendaftarkan *alarm* pada *broadcast receiver*. Sedangkan *AlarmManager* digunakan untuk mengatur kapan *alarm* akan diluncurkan.
3. Mengatur interval *alarm* yang akan diluncurkan setiap bulannya pada tanggal yang telah diatur pengasuh.
4. Menyimpan data *alarm* pada *database*.

```
tambahAlarm()
1: initialize datePicker
2: initialize calendar
3: get month from datePicker
4: get date from datePicker
5: set calendar month
6: set calendar date
7: initialize PendingIntent
8: initialize AlarmManager
9: call getInterval() function
10: set repeating alarm with interval
11: save alarm data to database
```

**Gambar 4.11 Pseudocode Implementasi
Mendaftarkan Pengingat Pemeriksaan Kesehatan**

```
getInterval()
1: month++
2: set calendar month
3: if month is December
4:   set calendar year+1
5: end if
6: if date is 31
7:   initialize maxDay
8:   set maxDay to calendar maximum day of
   the month
```

```

10: end if
11: else if date is 30
12:   if month is february
13:     set maxDay to calendr
        maximum day of the month
14:     set calendar day to maxDay
15:   end if
16:   else
17:     set calendar day to date
18:   end if
19: end if
20: else
21:   set calendar day to date
22: end if
23: initialize interval
24: set interval to calendar get time
25: return interval

```

Gambar 4.12 *Pseudocode* Mendapatkan Interval Waktu Pemeriksaan Kesehatan

4.2.11. Implementasi Mengatur Durasi Konfirmasi Keadaan

Gambar 4.13 merupakan *pseudocode* mengatur durasi konfirmasi keadaan kesehatan lansia. Pengasuh dapat memilih durasi yang telah ditetapkan oleh aplikasi yaitu 1 kali, 2 kali, atau 3 kali. Adapun langkah-langkah dalam mendaftarkan waktu pemeriksaan kesehatan pada *alarm* adalah sebagai berikut:

1. Pertama adalah dengan menginisialisasi *spinner*. Kemudian *spinner* akan diisi dengan *array of string* sebagai pilihannya. Setelah pengasuh memilih durasi yang diinginkan melalui *dropdown menu* maka *alarm* akan didaftarkan.
2. Durasi 1 kali akan mengaktifkan *alarm* pada pukul 10.00. Durasi dua kali akan mengaktifkan *alarm* pada pukul 10.00 dan 15.00. Durasi tiga kali akan mengaktifkan *alarm* pada pukul 10.00, 15.00, dan 19.00. *Alarm* akan diluncurkan setiap hari.

```

setDurationAlarm()
1: initialize spinner dropdown
2: initialize array of string pilihan
3: create array adapter for dropdown
4: if dropdown selection is 1 kali
5:     set alarm to fire at 10am
6: end if
7: else if dropdown selection is 2 kali
8:     set alarm to fire at 10am
9:     set alarm to fire at 3pm
10: end if
11: else
12:     set alarm to fire at 10am
13:     set alarm to fire at 3pm
14:     set alarm to fire at 7pm
15: end if

```

Gambar 4.13 Pseudocode Implementasi Mengatur Durasi Konfirmasi Keadaan

4.2.12. Implementasi *Text-to-Speech* untuk Notifikasi

Gambar 4.14 merupakan *pseudocode* memanfaatkan *text-to-speech* untuk pembacaan notifikasi bagi lansia. *Text-to-speech* akan membacakan notifikasi yang tertulis pada layar pengingat dan akan mengulang pembacaan terus hingga lansia mematikan pengingat. Agar pembacaan terus berulang maka dibutuhkan *utterance*. Adapun langkah-langkah pembacaan notifikasi dengan *text-to-speech* adalah sebagai berikut:

1. Pertama adalah dengan menginisialisasi objek *text-to-speech* serta *onInitListener*-nya, dan inisialisasi *utteranceProgressListener*, dan juga inisialisasi *TextView* yang akan dibacakan pada fungsi *onCreate()*.
2. Mengambil teks dari *TextView*.
3. Mengatur bahasa yang digunakan untuk pembacaan pada *onInitListener*.
4. Membuat fungsi *speak* untuk pembacaan *text-to-speech*.

5. Melakukan *override* fungsi ***onDone()*** pada *utteranceProgressListener* dengan memanggil fungsi *speak*.

```

onCreate()
1: initialize textToSpeechObject
2: initialize ttsOnInitListener
3: initialize TextView
4: initialize utteranceProgressListener
5: get string text from TextView

ttsOnInitListener()
1: set language for textToSpeechObject to IDN
2: call speak(text) function

speak(text)
1: initialize HashMap
2: put utteranceId to HashMap
3: call textToSpeechObject call function
   using text, QUEUE_ADD, and HashMap

utteranceProgressListener()
1: override onDone()
2: call speak(text) function

```

Gambar 4.14 Pseudocode Implementasi Text-to-Speech untuk Notifikasi

4.2.13. Implementasi Melakukan Panggilan Darurat dengan *SpeechRecognition*

Gambar 4.15 merupakan *pseudocode* melakukan panggilan darurat dengan memanfaatkan *speechRecognition*. Lansia dapat melakukan panggilan darurat dengan mengucapkan kata kunci yaitu “*help*” atau “*tolong*”. Pendeteksian suara dilakukan sebagai *background service*. Adapun langkah-langkah dalam melakukan panggilan darurat dengan *speechRecognition* adalah sebagai berikut:

1. Pertama adalah dengan menginisialisasi *speechRecognition* pada *service*, yaitu pada fungsi *onCreate()*. Kemudian dibuat objek *speechRecognition*.
2. Mengatur *speechRecognition listener*.
3. Menginisialisasi *speechRecognition intent* dengan *ACTION_RECOGNIZE_SPEECH*.
4. Memulai pendeteksian suara.
5. Apabila pengguna mengucapkan kata kunci berupa “*help*” atau “*tolong*” maka aplikasi akan memanggil nomor telepon yang telah dimasukkan sebelumnya.

```

speechRecognitionService()
1: initialize speechRecognition
2: set recognitionListener
3: initialize speechRecognitionIntent to
   ACTION_RECOGNIZE_SPEECH
4: start listening speechRecognition

onResults()
1: set ArrayList String to the result of
   SpeechRecognizer.RESULTS_RECOGNITION
2: if ArrayList String get first index equals
   help or tolong
3:   create intent action call
4:   set data call number
5:   add FLAG_ACTIVITY_NEW_TASK
6:   add FLAG_FROM_BACKGROUND
7:   start intent
8: end if
9: else
10:  start listening speechRecognition
11: end if

```

Gambar 4.15 Pseudocode Implementasi Melakukan Panggilan Darurat dengan *SpeechRecognition*

4.2.14. Implementasi Pengiriman Pesan Singkat Kepada Pengasuh

Gambar 4.16 merupakan *pseudocode* melakukan pengiriman pesan singkat kepada pengasuh lansia. Pesan singkat

akan dikirimkan apabila lansia terdeteksi terjatuh dan tidak melakukan konfirmasi, serta saat lansia dalam keadaan kesehatan tidak baik. Adapun langkah-langkah dalam pengiriman pesan singkat kepada pengasuh adalah sebagai berikut:

1. Inisialisasi *SmsManager* dan membuat objeknya dengan memanggil fungsi ***getDefault()***.
2. Inisialisasi *latitude* dan *longitude*. *Latitude* dan *longitude* akan diambil dari *location service* yang akan terus memperbarui nilainya selama perangkat hidup.
3. Inisialisasi *string* pesan dan menentukan pesan yang akan dikirimkan kepada pengasuh.
4. Inisialisasi *uri* dan memasukkan lokasi *latitude* dan *longitude* untuk tautan Google Maps.
5. Memanggil fungsi ***sendTextMessage()*** dengan parameter *destination address*, *service center address*, *text*, dan *sent intent*. Atur parameter dengan nomor telepon pengasuh, *null*, pesan, *null*.

```

sendSMS ()
1:  initialize SmsManager
2:  get default SmsManager
3:  initialize latitude
4:  initialize longitude
5:  initialize string textMessage
6:  initialize uri googleMaps
7:  get latitude from location service
8:  get longitude from location service
9:  add latitude and longitude to uri
10: call sendTextMessage function from
    SmsManager

```

Gambar 4.16 Pseudocode Implementasi Pengiriman Pesan Singkat Kepada Pengasuh

4.3 Implementasi Antarmuka

Pada subbab ini akan dijelaskan mengenai hasil implementasi antarmuka aplikasi *fall detection* yang telah dilakukan perancangannya pada Bab III.

4.3.1. Implementasi Antarmuka Menu Awal

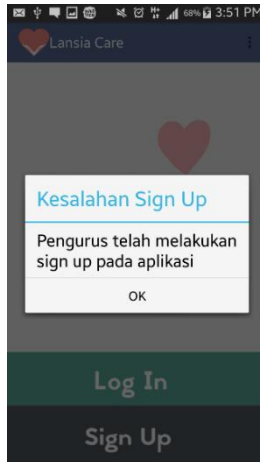
Gambar 4.17 merupakan implementasi antarmuka menu awal saat aplikasi baru dibuka. Pada antarmuka menu awal terdapat pilihan *log in* dan *sign up*.

Pilihan *log in* digunakan untuk masuk ke menu *log in*, sedangkan pilihan *sign up* digunakan untuk masuk ke menu *sign up*, namun sebelum *sign up* akan dilakukan pengecekan apakah aplikasi telah dilakukan *sign up* sebelumnya.



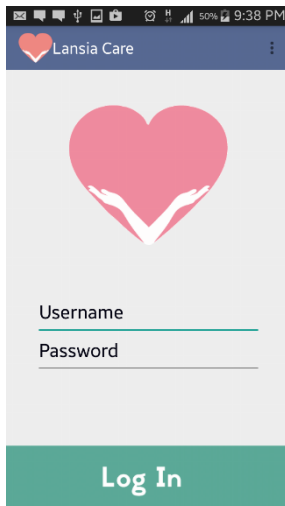
Gambar 4.17 Implementasi Antarmuka Menu Awal

Pada saat *sign up* akan dilakukan pengecekan apakah pengasuh sudah pernah membuat akun, hal ini dikarenakan pengaturan pada aplikasi hanya dapat dilakukan oleh satu pengasuh saja. Apabila telah terdapat akun, namun pilihan *sign up* dipilih akan muncul *alert dialog* kesalahan seperti pada Gambar 4.18.



Gambar 4.18 Implementasi Alert Dialog Kesalahan Sign Up

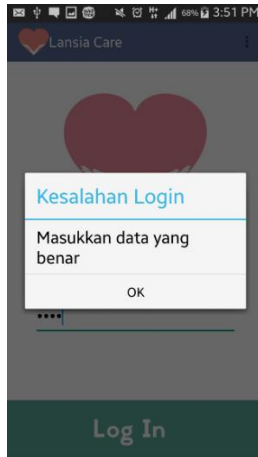
4.3.2. Implementasi Antarmuka Menu Login



Gambar 4.19 Implementasi Antarmuka Menu Log In

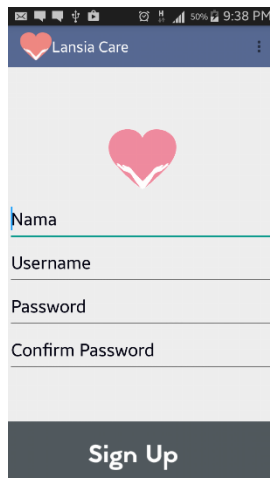
Gambar 4.19 merupakan implementasi antarmuka menu *log in*. Terdapat pilihan *log in* untuk masuk ke aplikasi. Apabila

username dan *password* cocok, maka pengasuh dapat masuk ke aplikasi. Apabila salah, maka akan muncul *alert dialog* kesalahan seperti ditunjukkan pada Gambar 4.20.



Gambar 4.20 Implementasi *Alert Dialog* Kesalahan *Log In*

4.3.3. Implementasi Antarmuka Menu *Sign Up*



Gambar 4.21 Implementasi Antarmuka Menu *Sign Up*

Gambar 4.21 merupakan implementasi antarmuka menu *sign up*. Terdapat pilihan *sign up* untuk mendaftarkan akun pengasuh. Saat pilihan *sign up* dipilih maka akan dilakukan pengecekan apakah *password* yang dimasukkan sudah sama dengan *confirm password*. Apabila sama, maka akun akan dibuat, apabila tidak sama maka akan muncul *alert dialog* kesalahan.

4.3.4. Implementasi Antarmuka Menu Utama



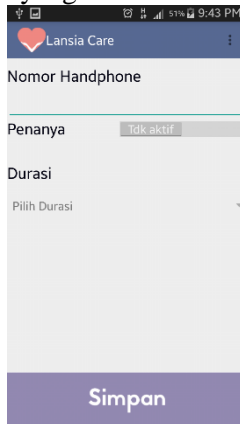
Gambar 4.22 Implementasi Antarmuka Menu Utama

Gambar 4.22 merupakan implementasi antarmuka menu utama yang muncul setelah pengasuh berhasil *log in* ke aplikasi. Terdapat empat pilihan pada menu ini, yaitu pengaturan, atur jadwal, dan ubah profil. Tiap-tiap pilihan akan masuk ke menu selanjutnya.

4.3.5. Implementasi Antarmuka Menu Pengaturan

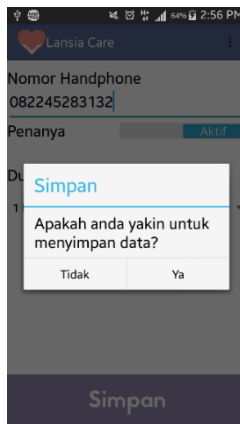
Gambar 4.23 merupakan implementasi antarmuka menu pengaturan. Terdapat *editText* nomor telepon untuk memasukkan nomor telepon pengasuh serta *switch* untuk mengaktifkan fitur

konfirmasi keadaan kesehatan lansia. Apabila *switch* diaktifkan maka *spinner dropdown* akan menampilkan pilihan durasi yang dapat dipilih oleh pengasuh. Terdapat pilihan simpan untuk menyimpan pengaturan yang telah dilakukan pengasuh.



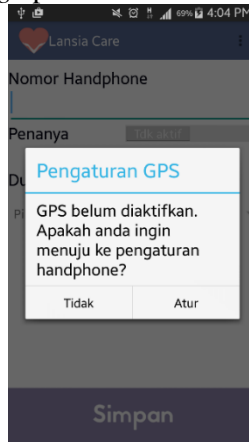
Gambar 4.23 Implementasi Antarmuka Menu Pengaturan

Saat pengasuh melakukan penyimpanan pengaturan akan muncul *alert dialog* konfirmasi penyimpanan seperti pada Gambar 4.24.



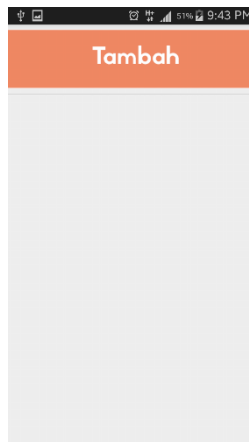
Gambar 4.24 Implementasi Alert Dialog Simpan Pengaturan

Selain itu, apabila GPS pada *smartphone* belum diaktifkan maka akan muncul *alert dialog* untuk menyalakan GPS dan akan masuk ke menu *setting* apabila memilih atur.

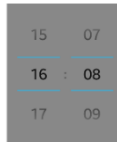


Gambar 4.25 Implementasi Alert Dialog Pengaturan GPS

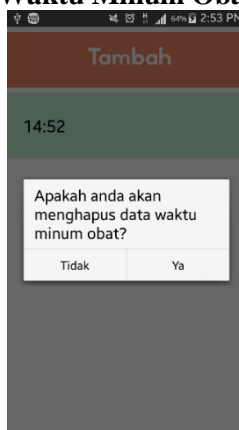
4.3.6. Implementasi Antarmuka Menu Atur Obat



Gambar 4.26 Implementasi Antarmuka Menu Atur Obat



Gambar 4.27 Implementasi Antarmuka Menu Tambah Waktu Minum Obat



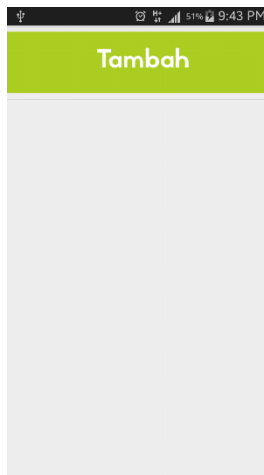
Gambar 4.28 Implementasi *Alert Dialog* Penghapusan Data Waktu Minum Obat

Gambar 4.26 merupakan implementasi antarmuka menu pengaturan waktu minum obat. Daftar waktu minum obat akan ditampilkan pada menu ini. Apabila pengasuh menekan waktu minum obat pada daftar akan tampil menu atur waktu minum obat. Apabila pengasuh menekan waktu minum obat cukup lama akan tampil *alert dialog* konfirmasi penghapusan waktu minum obat

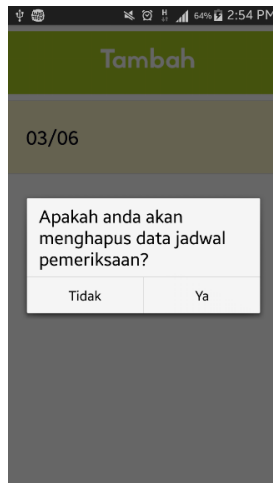
yang ditekan. Pengasuh dapat menambah waktu minum obat dengan memilih pilihan tambah pada tombol yang terletak di atas.

4.3.7. Implementasi Antarmuka Menu Atur Jadwal

Gambar 4.29 merupakan implementasi antarmuka menu pengaturan jadwal pemeriksaan kesehatan. Daftar waktu pemeriksaan kesehatan akan ditampilkan pada menu ini. Apabila pengasuh menekan waktu pemeriksaan kesehatan pada daftar akan tampil menu atur waktu pemeriksaan kesehatan. Apabila pengasuh menekan waktu pemeriksaan kesehatan cukup lama akan tampil *alert dialog* konfirmasi penghapusan waktu pemeriksaan kesehatan yang ditekan. Pengasuh dapat menambah waktu pemeriksaan kesehatan dengan memilih pilihan tambah pada tombol yang terletak di atas.

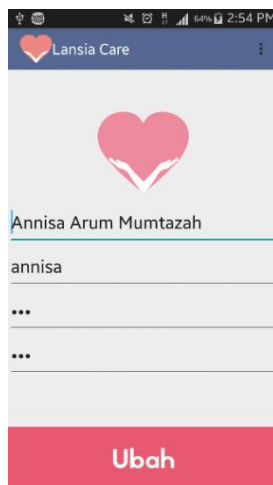


Gambar 4.29 Implementasi Antarmuka Menu Atur Jadwal



Gambar 4.30 Implementasi *Alert Dialog* Penghapusan Data Jadwal Pemeriksaan

4.3.8. Implementasi Antarmuka Menu Ubah Profil

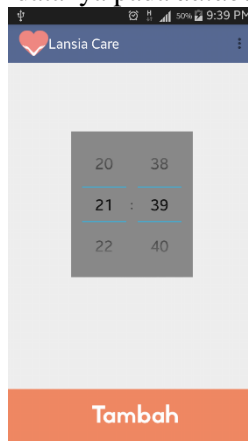


Gambar 4.31 Implementasi Antarmuka Menu Ubah Profil

Gambar 4.31 merupakan implementasi antarmuka menu ubah profil pengasuh. Terdapat *editText* nama, *username*, *password*, dan *confirm password* yang telah terisi dengan data pengasuh yang telah dimasukkan saat *sign up*. Terdapat pilihan ubah untuk menyimpan perubahan profil, namun sebelumnya akan dilakukan pengecekan apakah *password* dan *confirm password*-nya cocok.

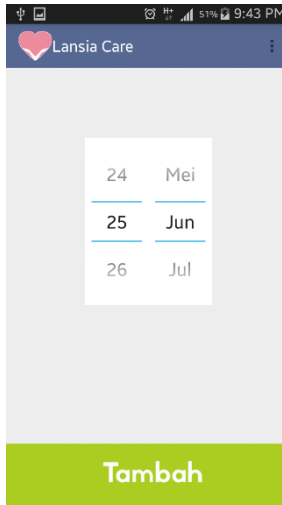
4.3.9. Implementasi Antarmuka Menu Tambah Waktu Minum Obat

Gambar 4.32 merupakan implementasi antarmuka menu tambah waktu minum obat. Terdapat *timePicker* untuk memilih jam dan menit yang diinginkan bagi lansia untuk minum obat. Terdapat pilihan tambah untuk mendaftarkan waktu tersebut ke *alarm* dan menyimpan datanya pada *database*.



Gambar 4.32 Implementasi Antarmuka Menu Tambah Waktu Minum Obat

4.3.10. Implementasi Antarmuka Menu Tambah Waktu Pemeriksaan Kesehatan



Gambar 4.33 Implementasi Antarmuka Menu Tambah Waktu Pemeriksaan Kesehatan

Gambar 4.33 merupakan implementasi antarmuka menu tambah waktu pemeriksaan kesehatan. Terdapat *datePicker* untuk memilih tanggal dan bulan yang diinginkan bagi lansia untuk memeriksakan kesehatannya. Terdapat pilihan tambah untuk mendaftarkan waktu tersebut ke *alarm* dan menyimpan datanya pada *database*.

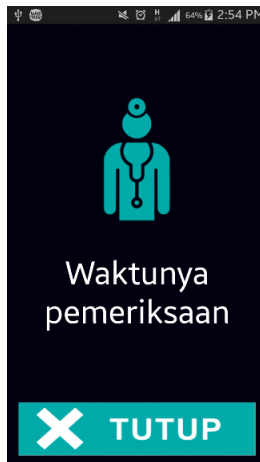
4.3.11. Implementasi Antarmuka Notifikasi Waktu Minum Obat

Gambar 4.34 merupakan implementasi antarmuka notifikasi yang muncul untuk mengingatkan lansia waktunya minum obat. Terdapat *textView* teks pengingat waktu minum obat yang akan disuarakan oleh *text-to-speech*, selain itu terdapat pilihan tutup untuk menutup notifikasi.



Gambar 4.34 Implementasi Antarmuka Notifikasi Waktu Minum Obat

4.3.12. Implementasi Antarmuka Notifikasi Waktu Pemeriksaan Kesehatan



Gambar 4.35 Implementasi Antarmuka Notifikasi Waktu Pemeriksaan Kesehatan

Gambar 4.35 merupakan implementasi antarmuka notifikasi yang muncul untuk mengingatkan lansia waktunya periksa kesehatan. Terdapat *textView* teks pengingat waktu periksa kesehatan yang akan disuarakan oleh *text-to-speech*, selain itu terdapat pilihan tutup untuk menutup notifikasi.

4.3.13. Implementasi Antarmuka Notifikasi Konfirmasi Keadaan

Gambar 4.36 merupakan implementasi antarmuka notifikasi yang muncul untuk menanyakan keadaan kesehatan lansia saat itu. Terdapat *textView* teks konfirmasi keadaan kesehatan yang akan disuarakan oleh *text-to-speech*, selain itu terdapat dua tombol geser untuk menjawab keadaan baik dan tidak baik. Apabila lansia menjawab keadaan kesehatannya tidak baik, maka aplikasi akan mengirim pesan singkat kepada pengasuh, apabila keadaan baik, maka notifikasi akan tertutup secara otomatis.



Gambar 4.36 Implementasi Antarmuka Notifikasi Konfirmasi Keadaan

4.3.14. Implementasi Antarmuka Notifikasi Konfirmasi Terjatuh

Gambar 4.37 merupakan implementasi antarmuka notifikasi yang muncul untuk mengkonfirmasi apakah lansia terjatuh setelah lansia terdeteksi terjatuh oleh aplikasi, hal ini adalah untuk menghindari *false positive* dari deteksi jatuh aplikasi. Terdapat *textView* teks konfirmasi terjatuh yang akan disuarakan oleh *text-to-speech*, selain itu terdapat dua tombol geser untuk menjawab tidak jatuh dan jatuh. Apabila lansia menjawab baik, maka aplikasi akan mengirim pesan singkat kepada pengasuh, apabila keadaan baik, maka notifikasi akan tertutup secara otomatis. Selain itu, terdapat *countdown timer* selama 10 detik, apabila pengguna tidak melakukan konfirmasi dalam 10 detik, maka aplikasi akan langsung mengirim pesan singkat kepada pengasuh dan notifikasi akan menutup secara otomatis.



Gambar 4.37 Implementasi Antarmuka Notifikasi Konfirmasi Terjatuh

BAB V

UJI COBA DAN EVALUASI

5.1 Lingkungan Uji Coba Aplikasi

Dalam proses pengujian aplikasi dibutuhkan lingkungan uji coba yang disesuaikan standar kebutuhan. Uji coba aplikasi ini dilakukan dengan menggunakan satu buah *smartphone*. Adapun spesifikasi *smartphone* yang digunakan adalah sebagai berikut:

- Perangkat *mobile* Samsung Galaxy Note II GT-N7100
 - Sistem Operasi Android versi 4.4.2
 - *Processor* Quad Core 1.6GHz
 - *Memory Card Slot* microSD up to 64GB
 - *Internal memory* 16GB
 - 2 GB RAM
 - Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi Direct, DLNA, hotspot

5.2 Skenario Uji Coba

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan dalam tiga tahap yaitu pengujian fungsionalitas, pengujian kegunaan, dan pengujian akurasi deteksi jatuh. Pengujian fungsionalitas dilakukan dengan menggunakan metode kotak hitam (*black box*) yang menekankan pada hasil keluaran aplikasi. Pengujian kegunaan dilakukan dengan menggunakan metode kuisioner dan responden pada lansia dan pengasuh lansia. Pengujian akurasi dilakukan untuk menguji respon aplikasi terhadap beberapa aktivitas lansia.

5.2.1. Uji Coba Fungsionalitas

Pada uji coba fungsionalitas akan diuji semua fungsi yang telah dibuat pada program sesuai dengan perencanaan sistem aplikasi. Uji coba fungsionalitas meliputi semua kasus penggunaan yang telah dijelaskan pada Bab III. Hasil uji coba ditunjukkan dengan *screenshot* dari perangkat *smartphone* Samsung Galaxy Note II GT-N7100 pada Lampiran B.

5.2.1.1 Uji Coba Menjalankan Aplikasi Lansia Care

Tabel 5.1 Hasil Uji Coba Menjalankan Aplikasi Lansia Care

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka aplikasi Lansia Care	Tekan ikon aplikasi pada beranda <i>smartphone</i>	Masuk menu awal aplikasi	Sesuai	Gambar B. 1	Diterima

5.2.1.2 Uji Coba *Log In* pada Aplikasi

Tabel 5.2 Hasil Uji Coba *Log In* pada Aplikasi

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka menu <i>log in</i>	Tekan pilihan <i>log in</i> pada menu awal aplikasi	Masuk menu <i>log in</i>	Sesuai	Gambar B. 2	Diterima
Memasukkan <i>username</i> pada <i>edit text</i>	Tekan <i>edit text username</i> hingga keluar papan tombol	<i>Username</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>password</i> pada <i>edit text</i>	Tekan <i>edit text password</i> hingga keluar papan tombol	<i>Password</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Masuk ke dalam aplikasi	Tekan pilihan <i>log in</i> pada menu <i>log in</i>	Membuka menu utama	Sesuai	Gambar B. 3	Diterima

Memasukkan <i>password</i> yang tidak sesuai	Tekan pilihan <i>log in</i> dengan <i>password</i> yang tidak sesuai	Menampilkan <i>alert dialog</i> kesalahan <i>log in</i>	Sesuai	Gambar B. 4	Diterima
Memasukkan <i>username</i> dan <i>password</i> yang belum terdaftar	Tekan pilihan <i>log in</i> dengan <i>username</i> dan <i>password</i> yang belum terdaftar	Menampilkan <i>alert dialog</i> kesalahan <i>log in</i>	Sesuai	Gambar B. 4	Diterima

5.2.1.3 Uji Coba *Sign Up* pada Aplikasi

Tabel 5.3 Hasil Uji Coba *Sign Up* pada Aplikasi

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka menu <i>sign up</i>	Tekan pilihan <i>sign up</i> pada menu awal aplikasi	Masuk menu <i>sign up</i>	Sesuai	Gambar B. 5	Diterima

Memasukkan nama pada <i>edit text</i>	Tekan <i>edit text</i> nama hingga keluar papan tombol	Nama tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>username</i> pada <i>edit text</i>	Tekan <i>edit text</i> <i>username</i> hingga keluar papan tombol	<i>Username</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>password</i> pada <i>edit text</i>	Tekan <i>edit text</i> <i>password</i> hingga keluar papan tombol	<i>Password</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>confirm password</i> pada <i>edit text</i>	Tekan <i>edit text</i> <i>confirm password</i> hingga keluar papan tombol	<i>Confirm password</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Mendaftarkan data akun pada aplikasi	Tekan pilihan <i>sign up</i> pada menu <i>sign up</i>	Masuk kembali ke menu awal aplikasi	Sesuai	-	Diterima

Memasukkan <i>password</i> dan <i>confirm password</i> yang tidak cocok	Tekan pilihan <i>sign up</i> pada menu <i>sign up</i> dengan <i>password</i> dan <i>confirm password</i> yang tidak cocok	Muncul <i>alert dialog password</i> dan <i>confirm password</i> tidak cocok	Sesuai	Gambar B. 6	Diterima
Mengosongkan salah satu <i>edit text</i>	Isi <i>edit text</i> dengan mengosongkan salah satunya dan tekan pilihan <i>sign up</i>	Muncul <i>alert dialog</i> masukkan data secara lengkap	Sesuai	Gambar B. 7	Diterima

5.2.1.4 Uji Coba Mengatur Waktu Minum Obat

Tabel 5.4 Hasil Uji Coba Mengatur Waktu Minum Obat

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka menu atur waktu minum obat	Tekan pilihan jadwal obat pada menu utama	Masuk menu atur waktu minum obat	Sesuai	Gambar B. 8	Diterima
Membuka menu tambah waktu minum obat	Tekan pilihan tambah pada menu atur waktu minum obat	Masuk menu tambah waktu minum obat	Sesuai	Gambar B. 9	Diterima
Menambahkan data waktu minum obat pada <i>database</i>	<ul style="list-style-type: none"> Tekan pilihan tambah pada menu atur waktu minum obat 	Data waktu minum obat tersimpan pada <i>database</i>	Sesuai	-	Diterima

	<ul style="list-style-type: none"> • Mengatur waktu yang diinginkan pada <i>time picker</i> • Memilih pilihan tambah 				
Mendaftarkan data waktu minum obat pada <i>alarm</i>	<ul style="list-style-type: none"> • Tekan pilihan tambah pada menu atur waktu minum obat • Mengatur waktu yang diinginkan pada <i>time picker</i> • Memilih pilihan tambah 	Waktu minum obat terdaftar pada <i>alarm</i>	Sesuai	-	Diterima

Menampilkan daftar waktu minum obat pada menu jadwal obat	Tekan pilihan jadwal obat pada menu utama	Daftar waktu minum obat ditampilkan pada <i>list view</i>	Sesuai	Gambar B. 10	Diterima
Mengubah data waktu minum obat	<ul style="list-style-type: none"> • Tekan waktu minum obat yang akan diubah pada <i>listview</i> • Mengatur waktu yang diinginkan pada <i>time picker</i> • Memilih pilihan ubah 	<ul style="list-style-type: none"> • Waktu minum obat baru terdaftar pada <i>alarm</i> • Data waktu minum obat diperbarui pada <i>database</i> 	Sesuai	Gambar B. 11	Diterima
Menghapus data waktu minum obat	<ul style="list-style-type: none"> • Tekan lama waktu minum obat yang akan 	<ul style="list-style-type: none"> • Waktu minum obat terhapus pada <i>database</i> 	Sesuai	Gambar B. 12	Diterima

	dihapus pada <i>listview</i> • Pilih pilihan ya pada <i>alert dialog</i> penghapusan yang muncul	• <i>Alarm</i> dibatalkan pada waktu yang dihapus			
--	---	---	--	--	--

5.2.1.5 Uji Coba Mengatur Jadwal Pemeriksaan Kesehatan

Tabel 5.5 Hasil Uji Coba Mengatur Jadwal Pemeriksaan Kesehatan

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka menu atur waktu pemeriksaan kesehatan	Tekan pilihan jadwal periksa pada menu utama	Masuk menu atur waktu periksa kesehatan	Sesuai	Gambar B. 13	Diterima
Membuka menu tambah	Tekan pilihan tambah pada menu atur	Masuk menu tambah jadwal pemeriksaan	Sesuai	Gambar B. 14	Diterima

jadwal pemeriksaan	waktu pemeriksaan kesehatan				
Menambahkan data waktu pemeriksaan kesehatan pada <i>database</i>	<ul style="list-style-type: none"> • Tekan pilihan tambah pada menu atur jadwal periksa kesehatan • Mengatur tanggal yang diinginkan pada <i>date picker</i> • Memilih pilihan tambah 	Data waktu periksa kesehatan tersimpan pada <i>database</i>	Sesuai	-	Diterima
Mendaftarkan data waktu pemeriksaan	<ul style="list-style-type: none"> • Tekan pilihan tambah pada menu atur 	Waktu periksa kesehatan terdaftar pada <i>alarm</i>	Sesuai	-	Diterima

kesehatan pada <i>alarm</i>	jadwal Periksa kesehatan <ul style="list-style-type: none"> • Mengatur tanggal yang diinginkan pada <i>date picker</i> • Memilih pilihan tambah 				
Menampilkan daftar waktu pemeriksaan kesehatan pada menu jadwal Periksa kesehatan	Tekan pilihan jadwal Periksa pada menu utama	Daftar waktu Periksa kesehatan ditampilkan pada <i>list view</i>	Sesuai	Gambar B. 15	Diterima
Mengubah data waktu pemeriksaan kesehatan	<ul style="list-style-type: none"> • Tekan waktu Periksa kesehatan yang akan 	<ul style="list-style-type: none"> • Waktu Periksa kesehatan baru 	Sesuai	Gambar B. 16	Diterima

	diubah pada <i>listview</i> <ul style="list-style-type: none"> • Mengatur tanggal yang diinginkan pada <i>date picker</i> • Memilih pilihan ubah 	terdaftar pada <i>alarm</i> <ul style="list-style-type: none"> • Data waktu periksa kesehatan diperbarui pada <i>database</i> 			
Menghapus data waktu pemeriksaan kesehatan	<ul style="list-style-type: none"> • Tekan lama waktu periksa kesehatan yang akan dihapus pada <i>listview</i> • Pilih pilihan ya pada <i>alert dialog</i> penghapusan yang muncul 	<ul style="list-style-type: none"> • Waktu minum obat terhapus pada <i>database</i> • <i>Alarm</i> dibatalkan pada tanggal yang dihapus 	Sesuai	Gambar B. 17	Diterima

5.2.1.6 Uji Coba Ubah Profil Pengasuh

Tabel 5.6 Hasil Uji Coba Ubah Profil Pengasuh

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka menu ubah profil perawat	Tekan pilihan ubah profil pada menu utama	Masuk menu atur waktu pemeriksaan kesehatan	Sesuai	Gambar B. 18	Diterima
Memasukkan nama pada <i>edit text</i>	Tekan <i>edit text</i> nama hingga keluar papan tombol	Nama tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>username</i> pada <i>edit text</i>	Tekan <i>edit text</i> <i>username</i> hingga keluar papan tombol	<i>Username</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>password</i> pada <i>edit text</i>	Tekan <i>edit text</i> <i>password</i> hingga keluar papan tombol	<i>Password</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima

Memasukkan <i>confirm password</i> pada <i>edit text</i>	Tekan <i>edit text confirm password</i> hingga keluar papan tombol	<i>Confirm password</i> tertulis pada <i>edit text</i>	Sesuai	-	Diterima
Memasukkan <i>password</i> dan <i>confirm password</i> yang tidak cocok	Tekan pilihan <i>sign up</i> pada menu <i>sign up</i> dengan <i>password</i> dan <i>confirm password</i> yang tidak cocok	Muncul <i>alert dialog password</i> dan <i>confirm password</i> tidak cocok	Sesuai	Gambar B. 19	Diterima
Mengosongkan salah satu <i>edit text</i>	Isi <i>edit text</i> dengan mengosongkan salah satunya dan tekan pilihan <i>sign up</i>	Muncul <i>alert dialog</i> masukkan data secara lengkap	Sesuai	Gambar B. 20	Diterima
Menampilkan data pengasuh sebelum	Tekan pilihan ubah profil	Data pengasuh sebelum	Sesuai	Gambar B. 18	Diterima

diubah pada <i>edit text</i>	pada menu utama	diubah tampil pada <i>edit text</i>			
Melakukan perubahan data pengasuh	Tekan pilihan ubah pada menu ubah profil	Data pengasuh yang telah diubah terbaru pada <i>database</i>	Sesuai	-	Diterima

5.2.1.7 Uji Coba Pengaturan Aplikasi

Tabel 5.7 Hasil Uji Coba Pengaturan Aplikasi

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Membuka menu pengaturan aplikasi	Tekan pilihan pengaturan pada menu utama	Masuk menu pengaturan aplikasi	Sesuai	Gambar B. 21	Diterima
Memasukkan nomor telepon	Tekan <i>edit text</i> nomor telepon	Nomor telepon	Sesuai	Gambar B. 22	Diterima

pengasuh pada <i>edit text</i>	hingga keluar papan tombol	tertulis pada <i>edit text</i>			
Mengaktifkan konfirmasi keadaan dengan <i>switcher</i>	Menggeser <i>switcher</i> hingga aktif yang terpilih	Konfirmasi keadaan diaktifkan	Sesuai	Gambar B. 22	Diterima
Menonaktifkan konfirmasi keadaan dengan <i>switcher</i>	Menggeser <i>switcher</i> hingga tidak aktif yang terpilih	Konfirmasi keadaan dinonaktifkan	Sesuai	-	Diterima
Memilih durasi pada <i>drop down</i>	<ul style="list-style-type: none"> • Tekan <i>drop down</i> menu sehingga ditampilkan pilihan durasi • Pilih durasi yang diinginkan pada <i>drop</i> 	Durasi konfirmasi keadaan terpilih	Sesuai	Gambar B. 22	Diterima

	<i>down</i> dengan menekannya				
Menyimpan pengaturan aplikasi	Tekan pilihan simpan dan pilih ya pada <i>alert dialog</i> yang muncul	Pengaturan tersimpan	Sesuai	Gambar B. 23	Diterima
Membatalkan penyimpanan pengaturan aplikasi	Tekan pilihan simpan dan pilih tidak pada <i>alert dialog</i> yang muncul	Kembali ke menu pengaturan		Gambar B. 23	
Mengosongkan <i>edit text</i> nomor telepon saat menyimpan	Kosongkan <i>edit text</i> nomor telepon lalu tekan pilihan simpan	Muncul <i>error text</i> nomor telepon harus diisi	Sesuai	Gambar B. 24	Diterima

5.2.1.8 Uji Coba Panggilan Darurat dengan *Speech Recognition*

Tabel 5.8 Hasil Uji Coba Panggilan Darurat dengan *Speech Recognition*

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Memulai <i>speech recognition</i>	Memilih pilihan pengaturan pada menu awal	<i>Speech recognition as service</i> diaktifkan	Sesuai	-	Diterima
Melakukan panggilan darurat dengan mengatakan “tolong”	Mendekatkan <i>smartphone</i> ke mulut dan katakan tolong	Panggilan darurat pada pengasuh dilakukan	Sesuai	Gambar B. 25	Diterima
Melakukan panggilan darurat dengan	Mendekatkan <i>smartphone</i> ke mulut dan katakan <i>help</i>	Panggilan darurat pada pengasuh dilakukan	Sesuai	Gambar B. 25	Diterima

mengatakan “ <i>help</i> ”					
Tidak melakukan panggilan darurat	Mendekatkan <i>smartphone</i> ke mulut namun berkata selain “ <i>help</i> ” dan “tolong”	Tidak melakukan panggilan darurat	Sesuai	--	Diterima

5.2.1.9 Uji Coba Memunculkan Notifikasi

Tabel 5.9 Hasil Uji Coba Memunculkan Notifikasi

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Memunculkan notifikasi pengingat waktu minum obat pada waktu yang	Prakondisi: Pengasuh telah mengatur waktu minum obat	Notifikasi pengingat waktu minum obat muncul	Sesuai	Gambar B. 26	Diterima

telah ditentukan					
Menutup notifikasi pengingat waktu minum obat	Tekan pilihan tutup pada notifikasi pengingat waktu minum obat	Notifikasi pengingat waktu minum obat tertutup	Sesuai	-	Diterima
Memunculkan notifikasi pengingat waktu pemeriksaan kesehatan pada waktu yang telah ditentukan	Prakondisi: Pengasuh telah mengatur waktu waktu pemeriksaan kesehatan	Notifikasi pengingat waktu pemeriksaan kesehatan muncul	Sesuai	Gambar B. 27	Diterima
Menutup notifikasi pengingat waktu	Tekan pilihan tutup pada notifikasi pengingat waktu	Notifikasi pengingat waktu pemeriksaan	Sesuai	-	Diterima

pemeriksaan kesehatan	pemeriksaan kesehatan	kesehatan tertutup			
Memunculkan notifikasi konfirmasi keadaan pada waktu yang telah ditentukan aplikasi	Prakondisi: Pengasuh telah mengatur durasi konfirmasi keadaan	Notifikasi konfirmasi keadaan muncul	Sesuai	Gambar B. 28	Diterima
Menutup notifikasi konfirmasi keadaan	Geser tombol geser baik apabila keadaan baik atau geser tombol geser kurang baik apabila keadaan kurang baik	Notifikasi konfirmasi keadaan tertutup	Sesuai	-	Diterima
Memunculkan notifikasi	Prakondisi:	Notifikasi konfirmasi	Sesuai	Gambar B. 29	Diterima

konfirmasi terjatuh	Lansia terdeteksi terjatuh	terjatuh muncul			
Menutup notifikasi konfirmasi terjatuh secara manual	Geser tombol geser baik apabila tidak terjatuh atau geser tombol geser kurang baik apabila terjatuh	Notifikasi konfirmasi terjatuh tertutup	Sesuai	-	Diterima
Menutup notifikasi konfirmasi terjatuh secara otomatis	Menunggu hingga <i>countdown</i> 10 detik selesai	Notifikasi konfirmasi terjatuh tertutup	Sesuai	-	Diterima

5.2.1.10 Uji Coba Pengiriman Pesan Singkat pada Pengasuh

Tabel 5.10 Hasil Uji Coba Pengiriman Pesan Singkat pada Pengasuh

Butir Uji	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Gambar Hasil	Kesimpulan
Mengirim notifikasi pesan singkat ketika lansia terjatuh	Prakondisi: Lansia terdeteksi terjatuh	SMS terkirim kepada pengasuh	Sesuai	Gambar B. 30	Diterima
Mengirim notifikasi pesan singkat ketika lansia dalam keadaan kurang baik	Prakondisi: Lansia memilih pilihan kurang baik pada notifikasi konfirmasi keadaan	SMS terkirim pada pengasuh	Sesuai	Gambar B. 30	Diterima

5.2.2. Uji Coba Kegunaan

Uji coba kegunaan dilakukan dengan melakukan metode kuisioner dan responden secara langsung pada lansia maupun pengasuh lansia. Uji coba kegunaan meliputi kesesuaian fitur aplikasi dengan kebutuhan lansia serta kemudahan pengasuh dalam menggunakan aplikasi. Pengujian dilakukan penulis dengan metode *blackbox testing*.

Uji coba ini dilaksanakan pada tanggal 5-6 Juni 2015 bertempat di:

- Nama tempat : UPTD Griya Wreda milik Dinas Sosial Pemerintah Kota Surabaya
- Alamat : Jalan Medokan Asri Barat X Blok N-19A Rungkut Surabaya 60295

5.2.2.1 Uji Coba Kegunaan pada Lansia

Uji coba kegunaan pada lansia dilakukan dengan mewawancarai lansia secara langsung untuk menjawab kuisioner yang telah dibuat. Rincian isi kuisioner bagi lansia dapat dilihat pada Tabel 5.11

Tabel 5.11 Kuisioner Uji Coba Kegunaan Pada Lansia

No	Pernyataan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1.	Menurut saya, fungsi panggilan darurat dengan <i>speechRecognition</i> sesuai dengan kebutuhan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2.	Menurut saya, fungsi deteksi jatuh sesuai dengan kebutuhan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Menurut saya, fungsi pengingat jadwal pemeriksaan kesehatan sesuai dengan kebutuhan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Menurut saya, fungsi pengingat waktu minum obat sesuai dengan kebutuhan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Menurut saya, fungsi konfirmasi keadaan kesehatan sesuai dengan kebutuhan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Menurut saya, tampilan antarmuka pengingat waktu minum obat jelas dan mudah dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Menurut saya, tampilan antarmuka pengingat jadwal pemeriksaan kesehatan jelas dan mudah dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Menurut saya, tampilan antarmuka konfirmasi jatuh jelas dan mudah dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	Menurut saya, tampilan antarmuka konfirmasi keadaan kesehatan jelas dan mudah dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Lansia yang diwawancarai adalah lansia yang masih mandiri atau lansia yang masih dapat melakukan kegiatan sehari-

hari secara independen (tidak *bedrest*). Sepuluh lansia yang diwawancarai tertera pada Tabel 5.12.

Tabel 5.12 Responden Lansia

Nama	Jenis Kelamin	Usia
Walimah	Perempuan	73 tahun
Damirah	Perempuan	62 tahun
Tukinah	Perempuan	70 tahun
Anjani	Perempuan	75 tahun
Tamidjo	Laki-Laki	75 tahun
Marman	Laki-Laki	73 tahun
Miun	Laki-Laki	61 tahun
Sularno	Laki-Laki	86 tahun
Badjuri	Laki-Laki	77 tahun
Senen	Laki-Laki	85 tahun

Tabel 5.13 Hasil Kuisiener Uji Coba Kegunaan pada Lansia

No	Pernyataan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1.	Menurut saya, fungsi panggilan darurat dengan <i>speechRecognition</i> sesuai dengan kebutuhan		60%	40%	
2.	Menurut saya, fungsi deteksi jatuh sesuai dengan kebutuhan	30%	50%	20%	
3.	Menurut saya, fungsi pengingat jadwal pemeriksaan kesehatan sesuai dengan kebutuhan		100%		
4.	Menurut saya, fungsi pengingat waktu		100%		

	minum obat sesuai dengan kebutuhan				
5.	Menurut saya, fungsi konfirmasi keadaan kesehatan sesuai dengan kebutuhan		80%	20%	
6.	Menurut saya, tampilan antarmuka pengingat waktu minum obat jelas dan mudah dipahami		90%	10%	
7.	Menurut saya, tampilan antarmuka pengingat jadwal pemeriksaan kesehatan jelas dan mudah dipahami		90%	10%	
8.	Menurut saya, tampilan antarmuka konfirmasi jatuh jelas dan mudah dipahami		50%	50%	
9.	Menurut saya, tampilan antarmuka konfirmasi keadaan kesehatan jelas dan mudah dipahami		50%	50%	

5.2.2.2 Uji Coba Kegunaan pada Pengasuh Lansia

Uji coba kegunaan pada pengasuh lansia dilakukan dengan mendemonstrasikan aplikasi Lansia *Care* dan mengisi kuisioner yang berkaitan dengan kemudahan menjalankan aplikasi dan kegunaan fitur bagi kondisi lansia. Pengujian dilakukan dengan memberikan kesempatan kepada pengasuh untuk mencoba sendiri aplikasi yang telah dikembangkan. Penulis hanya memberikan petunjuk alur menjalankan aplikasi.

Tabel 5.14 Kuisisioner Uji Coba Kegunaan pada Pengasuh Lansia

No	Pernyataan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1.	Menurut saya, mudah untuk mendaftarkan diri pada aplikasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	Menurut saya, mudah untuk masuk ke dalam aplikasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Menurut saya, mudah untuk menambah jadwal minum obat untuk lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Menurut saya, mudah untuk menambah jadwal pemeriksaan kesehatan untuk lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Menurut saya, mudah untuk melakukan pengubahan jadwal minum obat untuk lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Menurut saya, mudah untuk melakukan pengubahan jadwal pemeriksaan kesehatan untuk lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Menurut saya, mudah untuk memasukkan data nomor telepon	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Menurut saya, mudah untuk menghapus jadwal minum obat lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9.	Menurut saya, mudah untuk menghapus jadwal pemeriksaan kesehatan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Menurut saya, mudah untuk mengatur durasi konfirmasi keadaan kesehatan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	Menurut saya, mudah untuk mengubah profil pengasuh	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12.	Menurut saya, fungsi pengingat jadwal pemeriksaan kesehatan sesuai dengan kebutuhan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13.	Menurut saya, fungsi pengingat waktu minum obat sesuai dengan kebutuhan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.	Menurut saya, fungsi konfirmasi keadaan kesehatan lansia sesuai dengan kebutuhan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15.	Menurut saya, fungsi panggilan darurat dengan speechRecognition sesuai dengan kebutuhan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16.	Menurut saya, fungsi deteksi jatuh sesuai dengan kebutuhan lansia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Tabel 5.15 Responden Pengasuh

Nama	Jenis Kelamin	Usia
Zakharia	Laki-Laki	29 tahun
Ari Wati S.	Perempuan	26 tahun
Oki Sugiarto Nur Cahyono	Laki-Laki	29 tahun

Tabel 5.16 Hasil Kuisioner Uji Coba Kegunaan pada Pengasuh

No	Pernyataan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1.	Menurut saya, mudah untuk mendaftarkan diri pada aplikasi	33%	67%		
2.	Menurut saya, mudah untuk masuk ke dalam aplikasi		100%		
3.	Menurut saya, mudah untuk menambah jadwal minum obat untuk lansia		100%		
4.	Menurut saya, mudah untuk menambah jadwal pemeriksaan kesehatan untuk lansia		100%		
5.	Menurut saya, mudah untuk melakukan pengubahan jadwal minum obat untuk lansia		100%		
6.	Menurut saya, mudah untuk melakukan pengubahan jadwal pemeriksaan kesehatan untuk lansia		100%		

7.	Menurut saya, mudah untuk memasukkan data nomor telepon		100%		
8.	Menurut saya, mudah untuk menghapus jadwal minum obat lansia		100%		
9.	Menurut saya, mudah untuk menghapus jadwal pemeriksaan kesehatan lansia		100%		
10.	Menurut saya, mudah untuk mengatur durasi konfirmasi keadaan kesehatan lansia	33%	67%		
11.	Menurut saya, mudah untuk mengubah profil pengasuh		100%		
12.	Menurut saya, fungsi pengingat jadwal pemeriksaan kesehatan sesuai dengan kebutuhan lansia	67%	33%		
13.	Menurut saya, fungsi pengingat waktu minum obat sesuai dengan kebutuhan lansia	67%	33%		
14.	Menurut saya, fungsi konfirmasi keadaan kesehatan lansia sesuai dengan kebutuhan lansia	100%			
15.	Menurut saya, fungsi panggilan darurat dengan speechRecognition	67%	33%		

	sesuai dengan kebutuhan lansia				
16.	Menurut saya, fungsi deteksi jatuh sesuai dengan kebutuhan lansia	33%	67%		

5.2.3. Uji Coba Akurasi Deteksi Jatuh

Uji coba akurasi deteksi jatuh dilakukan untuk mengetahui tingkat akurasi klasifikasi *k-Nearest Neighbour* ditinjau dari beberapa aktivitas fisik. Aktivitas fisik yang akan diuji adalah aktivitas fisik pada saat terjatuh dan aktivitas fisik pada saat tidak terjatuh. Uji coba ini menggunakan penghitungan *true positive*, *true negative*, *false positive*, dan *false negative*. Uji coba aktivitas fisik dilakukan pada pengguna dengan tinggi badan 148 cm.

Uji coba dilakukan sebanyak 10 kali pada setiap aktivitas fisik. Adapun jenis uji coba kasus berdasarkan perhitungan *true positive*, *true negative*, *false positive*, dan *false negative* adalah sebagai berikut:

- **True Positive (TP)**
Uji coba dikatakan *true positive* ketika aktivitas jatuh terdeteksi oleh sistem sebagai aktivitas jatuh.
- **True Negative (TN)**
Uji coba dikatakan *true negative* ketika aktivitas tidak jatuh terdeteksi oleh sistem sebagai aktivitas tidak jatuh.
- **False Positive (FP)**
Uji coba dikatakan *false positive* ketika aktivitas tidak jatuh namun terdeteksi oleh sistem sebagai aktivitas jatuh.
- **False Negative (FN)**
Uji coba dikatakan *false negative* ketika aktivitas jatuh namun terdeteksi oleh sistem sebagai aktivitas tidak jatuh.

Aktivitas fisik yang akan dilakukan uji coba adalah sebagai berikut:

- Aktivitas jatuh ke depan (aktivitas pengguna dikatakan jatuh ke depan apabila pengguna jatuh dalam keadaan telungkup hingga berbaring menghadap bawah).
- Aktivitas jatuh ke belakang (aktivitas pengguna dikatakan jatuh ke belakang apabila pengguna jatuh menghadap ke atas hingga berbaring).
- Aktivitas jatuh vertikal ke kanan (aktivitas pengguna dikatakan jatuh vertikal ke kanan apabila pengguna jatuh ke arah samping kanan secara vertikal).
- Aktivitas jatuh vertikal ke kiri (aktivitas pengguna dikatakan jatuh vertikal ke kiri apabila pengguna jatuh ke arah samping kiri secara vertikal).
- Aktivitas berbaring.
- Aktivitas berjalan.
- Aktivitas berlari.
- Aktivitas melompat.
- Aktivitas duduk (aktivitas pengguna dikatakan duduk ketika pengguna dalam keadaan berdiri atau berjalan kemudian duduk).

Tabel 5.17 merupakan tabel uji coba perhitungan aktivitas fisik pengguna dengan parameter *true positive*, *false positive*, *true negative*, dan *false negative*. Data pada Tabel 5.17 akan dilakukan penghitungan sensitivitas atau *recall*, spesifisitas, dan presisi. Sensitivitas atau *recall* menghitung proporsi dari kondisi yang benar dari suatu percobaan. Spesifisitas menghitung proporsi dari kondisi yang salah dari suatu percobaan. Presisi menghitung fraksi dari *instance* yang relevan.

Tabel 5.17 Hasil Uji Coba Deteksi Jatuh pada Aktivitas Pengguna

No	Aktivitas	Banyak Uji	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)
1.	Jatuh ke depan	10	10	0	0	0
2.	Jatuh ke belakang	10	8	0	2	0
3.	Jatuh ke samping kanan	10	10	0	0	0
4.	Jatuh ke samping kiri	10	10	0	0	0
5.	Duduk	10	0	1	0	9
6.	Berjalan	10	0	0	0	10
7.	Berlari	10	0	0	0	10
8.	Melompat	10	0	0	0	10
9.	Berbaring	10	0	0	0	10
Jumlah			38	1	2	49

Dari Tabel 5.17 didapatkan nilai total *true positive* sebanyak 38 kejadian, *false positive* sebanyak 1 kejadian, *false negative* sebanyak 2 kejadian, dan *true negative* sebanyak 49 kejadian. Nilai sensitivitas, spesifisitas, dan presisi adalah sebagai berikut:

$$\begin{aligned}
 \text{Sensitivitas} &= \frac{TP}{TP + FN} \\
 &= \frac{38}{38 + 2} \\
 &= 0,95
 \end{aligned}$$

$$\text{Spesifisitas} = \frac{TN}{TN + FP}$$

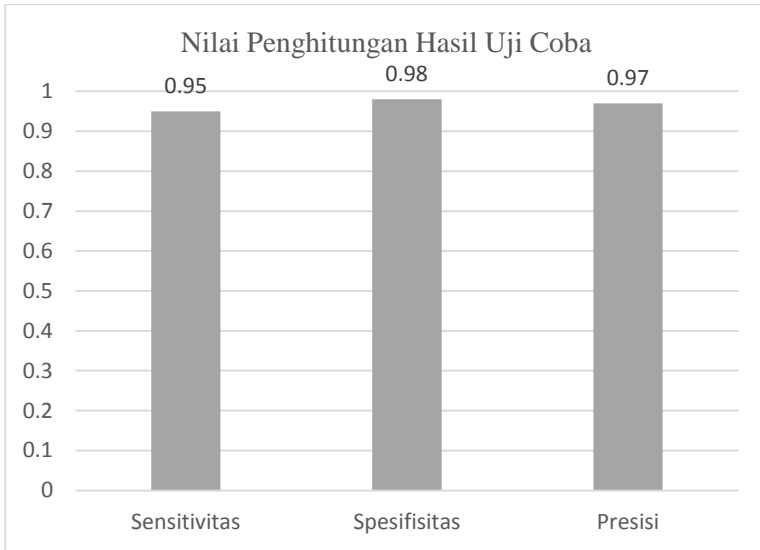
$$\begin{aligned}
 &= \frac{49}{49 + 1} \\
 &= 0,98
 \end{aligned}$$

$$\begin{aligned}
 \text{Presisi} &= \frac{TP}{TP + FP} \\
 &= \frac{38}{38 + 1} \\
 &= 0,97
 \end{aligned}$$

Berdasarkan informasi dalam Tabel 5.17 menunjukkan bahwa aktivitas jatuh ke depan, jatuh ke samping kanan, dan jatuh ke samping kiri memiliki nilai *true positive* sebesar 10 dan nilai persentase *false negative*, *true negative* dan *false positive* sebesar 0, yang artinya dalam 10 kali percobaan yang dilakukan terdeteksi 10 aktivitas terjatuh dan 0 kali tidak terdeteksi. Aktivitas jatuh ke belakang memiliki nilai *true positive* sebesar 8 dan nilai persentase *false negative* sebesar 2, serta *true negative* dan *false positive* sebesar 0, yang artinya aktivitas tersebut terdeteksi jatuh sebanyak 8 kali dan 2 kali tidak terdeteksi dari 10 kali percobaan.

Aktivitas fisik berjalan, berlari, melompat, dan berbaring memiliki nilai *true negative* sebesar 10 dan nilai *false positive*, *false negative*, dan *true positive* sebesar 0, yang berarti aktivitas tersebut terdeteksi 10 kali tidak terjatuh dari 10 kali percobaan. Sedangkan aktivitas duduk memiliki nilai *true negative* sebesar 9 dan nilai *false positive* sebesar 1 dan nilai *false negative* dan *true positive* sebesar 0, yang berarti dari 10 kali percobaan terdeteksi 9 kali tidak terjatuh dan 1 kali terdeteksi terjatuh.

Gambar 5.1 merupakan grafik nilai penghitungan sensitivitas, spesifisitas, dan presisi dari hasil uji coba aktivitas fisik pengguna. Sumbu *x* pada grafik menunjukkan jenis penghitungan akurasi, sedangkan sumbu *y* menunjukkan nilai penghitungan.



Gambar 5.1 Grafik Nilai Penghitungan Hasil Uji Coba

Gambar 5.1 menunjukkan bahwa nilai sensitivitas yang didapat adalah 0,95, nilai spesifisitas sebesar 0,98, dan nilai presisi sebesar 0,97. Sensitivitas yang tinggi yaitu 0, 95 mendekati 1 berarti kemungkinan aktivitas jatuh dideteksi terjatuh tinggi. Spesifisitas yang tinggi yaitu 0,98 mendekati 1 berarti kemungkinan aktivitas tidak jatuh dideteksi sebagai aktivitas tidak jatuh tinggi. Presisi yang tinggi yaitu 0,97 mendekati 1 berarti kemungkinan pendeteksian jatuh memang dari aktivitas jatuh tinggi.

5.3 Evaluasi

5.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.18. Berdasarkan data pada Tabel 5.18, semua fungsi yang diuji berhasil dan berjalan dengan baik. Sehingga dapat ditarik kesimpulan bahwa fungsionalitas aplikasi telah bekerja

sesuai dengan yang diharapkan dan menunjukkan proses dan hasil yang benar.

Tabel 5.18 Rangkuman Hasil Pengujian

Nama	Hasil
Pengujian menjalankan aplikasi	Berhasil
Pengujian <i>log in</i> pada aplikasi	Berhasil
Pengujian <i>sig up</i> pada aplikasi	Berhasil
Pengujian mengatur waktu minum obat	Berhasil
Pengujian mengatur jadwal pemeriksaan kesehatan	Berhasil
Pengujian ubah profil pengasuh	Berhasil
Pengujian pengaturan aplikasi	Berhasil
Pengujian panggilan darurat dengan <i>speech recognition</i>	Berhasil
Pengujian memunculkan notifikasi	Berhasil
Pengujian pengiriman pesan singkat pada pengasuh	Berhasil

5.3.2. Evaluasi Pengujian Kegunaan

Berdasarkan hasil kuisioner pada Tabel 5.13 dapat ditarik kesimpulan bahwa lansia menganggap bahwa aplikasi sesuai dengan kebutuhan mereka, ditunjukkan oleh 60% lansia setuju bahwa fungsi panggilan darurat dengan *speech recognition* sesuai dengan kebutuhannya, 80% lansia setuju bahwa fungsi deteksi jatuh sesuai dengan kebutuhannya, 100% lansia setuju bahwa fungsi pengingat waktu minum obat dan jadwal pemeriksaan kesehatan sesuai dengan kebutuhannya, dan 80% lansia setuju bahwa fungsi konfirmasi keadaan kesehatan sesuai dengan kebutuhannya.

Berdasarkan hasil kuisioner pada Tabel 5.16 dapat ditarik kesimpulan bahwa pengasuh lansia menganggap bahwa aplikasi mudah digunakan, ditunjukkan dengan 100% pengasuh lansia setuju bahwa aplikasi mudah digunakan. Selain itu, pengasuh juga menanggapi bahwa fungsionalitas aplikasi sesuai dengan kebutuhan lansia, ditunjukkan dengan 100% pengasuh lansia

menyatakan bahwa fungsionalitas aplikasi sesuai dengan kebutuhan lansia.

Dapat disimpulkan, pengoperasian aplikasi mudah dan fungsi-fungsi pada aplikasi sesuai dengan kebutuhan lansia.

5.3.3. Evaluasi Pengujian Akurasi

Berdasarkan hasil penghitungan sensitivitas dan spesifisitas pada uji coba akurasi deteksi jatuh didapat penghitungan nilai sensitivitas sebesar 0,95, nilai spesifisitas sebesar 0,98, dan nilai presisi sebesar 0,97. Nilai tersebut mendekati nilai 1. Sehingga dapat disimpulkan deteksi jatuh pada aplikasi memiliki akurasi yang baik.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap implementasi pada “Rancang Bangun Aplikasi Lansia *Care* untuk *Monitoring* Lansia yang Tinggal Sendiri Berbasis Android” diambil kesimpulan sebagai berikut:

1. Fungsionalitas aplikasi berjalan dengan baik dan sebagaimana mestinya.
2. Aplikasi berhasil memunculkan notifikasi waktu minum obat dan waktu pemeriksaan kesehatan pada waktu yang telah ditentukan dengan mengimplementasikan *text-to-speech*.
3. Aplikasi berhasil memunculkan notifikasi konfirmasi konfirmasi keadaan pada durasi waktu yang telah ditentukan dengan mengimplementasikan *text-to-speech*.
4. Aplikasi berhasil mendeteksi aktifitas fisik pengguna dengan nilai sensitivitas sebesar 0,95, nilai spesifisitas sebesar 0,98, dan nilai presisi sebesar 0,97 dengan melakukan pengolahan data sensor *accelerometer* dan *gyroscope* yang diklasifikasi dengan *k-Nearest Neighbour*.
5. Aplikasi berhasil memunculkan notifikasi konfirmasi terjatuh dengan mengimplementasikan *text-to-speech* serta pesan singkat ketika pengguna terdeteksi terjatuh.
6. Aplikasi berhasil mendapatkan posisi lansia saat ini menggunakan *location listener* yang diperbarui setiap 2 menit sekali.

7. Aplikasi berhasil melakukan panggilan darurat dengan *speechRecognizer* dengan menggunakan *Google Voice Command* yang berjalan pada *background process* ketika pengguna mengucapkan kata kunci yaitu “*help*” maupun “*tolong*”.
8. Aplikasi yang telah dibuat dapat dioperasikan dengan mudah oleh pengasuh berdasar hasil uji coba dimana 100% pengasuh menyatakan bahwa aplikasi mudah untuk dioperasikan.
9. Aplikasi yang telah dibuat telah sesuai dengan kebutuhan lansia.
10. Fungsionalitas aplikasi sesuai dengan kebutuhan lansia.

6.2. Saran

Saran yang diberikan untuk pengembangan “Rancang Bangun Aplikasi Lansia *Care* untuk *Monitoring* Lansia yang Tinggal Sendiri Berbasis Android” pada tugas akhir ini antara lain:

1. Menghilangkan bunyi pendeteksian *Google Voice Search* tanpa menghilangkan fitur *speaker*.
2. Menambah jumlah dan posisi *smartphone* ataupun sensor di beberapa titik tubuh pengguna agar dapat mendeteksi aktivitas fisik pengguna yang lebih bervariasi dan lebih akurat.
3. Menggunakan perbandingan algoritma selain *k-Nearest Neighbour* agar akurasi pendeteksian semakin meningkat.
4. Memperbaiki desain tampilan antarmuka agar lebih sesuai dengan kebutuhan pengguna.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap implementasi pada “Rancang Bangun Aplikasi Lansia *Care* untuk *Monitoring* Lansia yang Tinggal Sendiri Berbasis Android” diambil kesimpulan sebagai berikut:

1. Fungsionalitas aplikasi berjalan dengan baik dan sebagaimana mestinya.
2. Aplikasi berhasil memunculkan notifikasi waktu minum obat dan waktu pemeriksaan kesehatan pada waktu yang telah ditentukan dengan mengimplementasikan *text-to-speech*.
3. Aplikasi berhasil memunculkan notifikasi konfirmasi konfirmasi keadaan pada durasi waktu yang telah ditentukan dengan mengimplementasikan *text-to-speech*.
4. Aplikasi berhasil mendeteksi aktifitas fisik pengguna dengan nilai sensitivitas sebesar 0,95, nilai spesifisitas sebesar 0,98, dan nilai presisi sebesar 0,97 dengan melakukan pengolahan data sensor *accelerometer* dan *gyroscope* yang diklasifikasi dengan *k-Nearest Neighbour*.
5. Aplikasi berhasil memunculkan notifikasi konfirmasi terjatuh dengan mengimplementasikan *text-to-speech* serta pesan singkat ketika pengguna terdeteksi terjatuh.
6. Aplikasi berhasil mendapatkan posisi lansia saat ini menggunakan *location listener* yang diperbarui setiap 2 menit sekali.

7. Aplikasi berhasil melakukan panggilan darurat dengan *speechRecognizer* dengan menggunakan *Google Voice Command* yang berjalan pada *background process* ketika pengguna mengucapkan kata kunci yaitu “*help*” maupun “*tolong*”.
8. Aplikasi yang telah dibuat dapat dioperasikan dengan mudah oleh pengasuh berdasar hasil uji coba dimana 100% pengasuh menyatakan bahwa aplikasi mudah untuk dioperasikan.
9. Aplikasi yang telah dibuat telah sesuai dengan kebutuhan lansia.
10. Fungsionalitas aplikasi sesuai dengan kebutuhan lansia.

6.2. Saran

Saran yang diberikan untuk pengembangan “Rancang Bangun Aplikasi Lansia *Care* untuk *Monitoring* Lansia yang Tinggal Sendiri Berbasis Android” pada tugas akhir ini antara lain:

1. Menghilangkan bunyi pendeteksian *Google Voice Search* tanpa menghilangkan fitur *speaker*.
2. Menambah jumlah dan posisi *smartphone* ataupun sensor di beberapa titik tubuh pengguna agar dapat mendeteksi aktivitas fisik pengguna yang lebih bervariasi dan lebih akurat.
3. Menggunakan perbandingan algoritma selain *k-Nearest Neighbour* agar akurasi pendeteksian semakin meningkat.
4. Memperbaiki desain tampilan antarmuka agar lebih sesuai dengan kebutuhan pengguna.

DAFTAR PUSTAKA

- [1] A. J. Pitoyo, E. Kiswanto, P. N. Rahmat and S. P. Lestariningsih, *Menjadi Lansia Idahaman: Sehat Fisik, Sehat Sosial Psikologis, dan Mandiri Ekonomi*, Jakarta: Direktorat Kerjasama Pendidikan Kependudukan BKKBN, 2013.
- [2] S. Tamher and Noorkasiani, *Kesehatan Usia Lanjut dengan Pendekatan Asuhan Keperawatan*, Jakarta: Salemba Medika, 2009.
- [3] Senior Homes, "Senior Homes," [Online]. Available: <http://www.seniorhomes.com/p/aging-issue-an-introduction-to-elder-care/>. [Accessed 3 December 2014].
- [4] J. Sutherland, "An Introduction to Sensors and Transducers," [Online]. Available: <http://www.mfg.mtu.edu/cyberman/machtool/machtool/sensors/intro.html>. [Accessed 22 May 2015].
- [5] Android Developer, "Sensors Overview," [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html. [Accessed 4 May 2015].
- [6] SENSR, "Practical Guide to Accelerometers," [Online]. Available: www.sensr.com. [Accessed 20 December 2014].
- [7] GSM Arena, "Sensors," [Online]. Available: <http://www.gsmarena.com/glossary.php3?term=sensors>. [Accessed 4 May 2015].
- [8] S. Kabai, "Gyroscope," Wolfram Demonstrations Project, 28 September 2007. [Online]. Available: <http://demonstrations.wolfram.com/Gyroscope/>. [Accessed 20 December 2014].
- [9] N. Ravi, N. Dandekar, P. Mysore and M. Littman, "Activity Recognition from Accelerometer Data," in *The Seventeenth*

Conference on Innovative Applications of Artificial Intelligence (IAAI), Pittsburgh, 2005.

- [10] S. Brahler, "Analysis of the Android Architecture," Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft, Karlsruhe, 2010.
- [11] Android Developer, "Android Fundamentals," [Online]. Available:
<http://developer.android.com/guide/components/fundamentals.html> diakses. [Accessed 2 May 2015].
- [12] Android Developer, "SensorEventListener," [Online]. Available:
<http://developer.android.com/reference/android/hardware/SensorEventListener.html>. [Accessed 2 May 2015].
- [13] Android Developer, "RecognitionListener," [Online]. Available:
<http://developer.android.com/reference/android/speech/RecognitionListener.html>. [Accessed 4 May 2015].
- [14] Android Developer, "LocationListener," [Online]. Available:
<http://developer.android.com/reference/android/location/LocationListener.html>. [Accessed 3 May 2015].
- [15] Android Developer, "SmsManager," [Online]. Available:
<http://developer.android.com/reference/android/telephony/SmsManager.html>. [Accessed 4 May 2015].
- [16] L. Sahu and A. Dhole, "Hindi & Telugu Text-to-Speech Synthesis (TTS) and inter-language text Conversion," *International Journal of Scientific and Research Publication*, vol. 2, p. 1, 2012.
- [17] M. A. Anusaya and S. K. Katti, "Speech Recognition by Machine: A Review," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 6, p. 181, 2009.

- [18] Electronic Communications Committee (ECC), "Short Message Service (SMS) in Fixed and Mobile Networks," Gothenburg, 2004.
- [19] GPS NAVSTAR, Global Positioning System Standard Positioning Service Signal Specification, USA: Navstar, 1995.
- [20] Wikipedia, "Training Set," [Online]. Available: http://en.wikipedia.org/wiki/Training_set. [Accessed 4 May 2015].
- [21] Wikipedia, "Average," [Online]. Available: <http://en.wikipedia.org/wiki/Average>. [Accessed 10 June 2015].
- [22] Wikipedia, "Simpangan Baku," [Online]. Available: http://id.wikipedia.org/wiki/Simpangan_baku. [Accessed 6 June 2015].
- [23] T. Abeel, Y. Van de Peer and Y. Saeys, "Java-ML: A Machine Learning Library," *Journal of Machine Learning Research* 10, p. 1, 2009.
- [24] R. J. Henery, "Classification," in *Machine Learning, Neural and Statistical Classification*, Strathclyde, 1994, p. 6.
- [25] O. Sutton, "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction," February 2012. [Online]. Available: www.math.le.ac.uk/people/ag153/homepage/KNN/OliverKNN_Talk.pdf. [Accessed 2 May 2015].

LAMPIRAN A

Tabel A. 1 Data *Training* Deteksi Jatuh

AccX	AccY	AccZ	StDevAX	StDevAY	StDevAZ	GyroX	GyroY	GyroZ	StDevGX	StDevGY	StDevGZ	class
1.45328	4.97563	-3.29011	3.3613	4.753562	4.69277	0.04922	0.486982	0.193812	0.584796	0.849865	0.425299	jatuh
2.588611	4.037582	-2.9214	4.172685	5.666545	4.127905	0.281334	0.182084	0.187719	0.794118	0.443488	0.547893	jatuh
2.876873	3.405034	-4.67205	3.723628	5.190079	5.39135	0.487562	0.121425	0.421558	1.918388	0.813014	1.704001	jatuh
1.521755	4.530308	-3.13257	4.229859	5.65586	6.105947	0.47762	-0.11099	0.565524	1.587956	1.509722	1.845396	jatuh
2.05614	4.321055	-3.09475	4.283106	4.903538	4.448842	0.155923	0.258304	0.152319	0.691692	0.570567	0.473569	jatuh
2.048479	4.797023	-3.88052	3.319666	5.264738	5.535388	-0.38452	0.006338	0.485989	2.351517	2.395986	1.905689	jatuh
1.041478	5.734113	-1.99389	4.491622	3.686124	5.279029	0.066019	0.396772	0.404332	0.824472	0.674886	0.896212	jatuh
2.478956	4.815218	-4.0395	5.262662	4.413352	5.583971	0.334204	0.253066	0.456836	1.273716	0.666793	1.28967	jatuh
1.884237	4.258327	-3.27718	4.189965	4.570513	5.602951	0.080314	0.263848	0.06805	0.729635	0.846442	0.441671	jatuh
2.322376	3.073676	-3.54533	4.439139	5.976828	5.191116	0.521404	-0.29891	0.407814	1.741932	1.657164	1.133622	jatuh
2.472253	4.780263	3.224032	4.557191	6.228706	4.029137	-0.24335	0.088759	0.423131	0.63703	0.903928	1.154456	jatuh
2.394681	5.518635	2.458366	5.55851	5.322401	4.017778	-0.14241	-0.0898	0.455507	0.578793	1.061521	1.470493	jatuh
4.296634	5.211698	2.231396	5.52167	6.794441	4.287103	-0.33007	-0.15014	0.558254	1.021242	0.586731	1.722014	jatuh
4.619373	4.12904	1.864125	7.078184	6.561552	2.457174	-0.19198	-0.10852	0.282663	0.541745	0.598732	0.971759	jatuh
4.270777	4.373249	2.659479	5.618324	6.110316	4.113058	-0.3299	-0.00104	0.283472	0.823788	0.532885	1.355072	jatuh
-1.97617	4.949773	4.716099	2.108579	5.809884	5.235078	-0.04134	-0.03474	-0.05024	1.440818	0.603513	0.639995	tidak jatuh
-2.35973	2.706406	5.684314	2.451114	6.629531	5.329368	0.05368	-0.09592	-0.10285	1.433387	0.852731	0.828652	tidak jatuh
-2.75333	2.965458	6.082709	2.448182	7.2528	5.545785	-0.18431	-0.25337	-0.34996	1.479291	0.841119	0.884458	tidak jatuh
-0.62584	9.399157	2.061408	1.039789	1.372304	2.01027	0.00617	0.37162	0.076709	0.711916	0.539076	0.321029	tidak jatuh
-0.82696	9.76547	1.586398	1.063542	1.558236	1.811117	-0.05253	0.789803	0.05565	0.469754	1.028036	0.184927	tidak jatuh
-0.90357	8.901163	1.430296	0.886785	1.003739	1.532227	0.057116	0.124968	-0.02654	0.463635	1.101458	0.299013	tidak jatuh

-1.1631	9.239703	1.459505	0.485781	1.086667	1.681628	-0.07256	0.001268	-0.00942	0.791479	0.949642	0.293266	tidak jatuh
-1.22679	8.503247	1.891898	2.735824	4.372445	1.47049	-0.06112	0.075808	-0.12709	1.307124	0.819924	0.560337	tidak jatuh
-1.38241	7.971255	1.797088	2.58717	5.57682	1.642786	-0.40416	0.585484	-0.0899	1.286337	1.519744	0.735697	tidak jatuh
-1.24882	6.951325	1.875618	2.630278	4.143177	1.321578	-0.09679	0.160474	0.163132	1.425941	1.417464	0.686672	tidak jatuh
-1.97713	8.561186	1.438915	3.336126	6.602365	1.693589	-0.012	-0.17107	0.007498	1.637075	1.886132	0.683318	tidak jatuh
-2.22517	11.01285	1.534204	4.691986	5.987052	2.706938	-0.78974	0.316291	-0.20351	1.286177	2.210342	0.776821	tidak jatuh
-2.37218	9.796595	1.885673	4.419685	5.415881	3.299402	-0.52342	0.543227	-0.24132	1.364135	1.486433	0.93014	tidak jatuh
-1.38768	7.446925	2.775359	1.921395	7.514943	4.35581	-0.13399	-0.10539	-0.18731	0.894024	0.542617	0.737272	tidak jatuh
-1.35081	7.377972	3.016215	2.443988	7.724267	4.581063	0.008949	-0.00702	-0.02735	0.982418	0.458623	0.777431	tidak jatuh
2.568021	1.304361	-8.36965	1.901755	0.610948	2.772439	0.064202	-0.19551	-0.02147	0.242123	0.631005	0.032373	tidak jatuh
2.216073	1.269885	-8.89829	1.442869	0.612296	1.152343	0.009911	-0.12961	-0.01232	0.119912	0.6089	0.0161	tidak jatuh
1.896208	1.122881	-9.15064	0.055495	0.021722	0.071024	-0.01469	0.003222	-0.01	0.019508	0.025451	0.010018	tidak jatuh
2.275449	1.404439	-8.88201	1.108549	0.781646	1.083392	-0.17017	0.193736	0.024771	0.320682	0.430704	0.062606	tidak jatuh
-6.48015	-0.47645	1.882321	6.613633	8.148744	4.665651	-0.16052	-0.11758	-0.18468	0.575337	1.535407	0.770904	jatuh
-4.34787	3.660734	0.935654	3.406097	5.837636	4.00784	0.056474	-0.04931	-0.35945	0.405213	0.572027	1.020472	jatuh
-4.39719	3.985388	-1.56293	5.14836	6.306515	4.562303	0.008338	0.389793	-0.73238	0.638771	1.773621	2.206184	jatuh
-5.22558	4.33973	0.081403	4.573151	4.820395	2.532717	0.02668	0.020052	-0.25459	0.379607	0.578153	0.691662	jatuh
-5.12263	3.994007	-2.19596	3.036859	5.481859	3.724084	0.149601	-0.23902	-0.30307	0.437024	1.63559	0.841805	jatuh
3.671269	2.788766	2.600582	6.687205	3.636832	6.112025	0.265024	-0.71693	-0.24389	2.687915	2.596672	1.68002	tidak jatuh
0.998861	5.789659	2.45262	5.178496	4.140735	4.885693	0.081367	-0.87963	-0.65783	1.613104	2.331645	1.586817	tidak jatuh

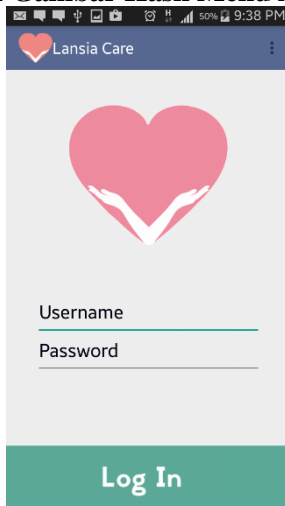
0.19393	4.907156	5.348168	4.468799	2.922573	4.638022	-0.0759	-0.35878	-0.5134	0.890373	2.05642	1.035103	tidak jatuh
0.360567	4.508761	6.977661	3.705098	2.485197	4.107187	0.066462	-0.05873	-0.2231	0.465377	1.851891	0.627463	tidak jatuh
-2.10307	0.606212	6.451655	1.628408	6.454651	3.199433	0.20041	0.138155	0.088911	0.778559	0.333009	0.432628	tidak jatuh
-8.31075	-0.07997	-4.50661	0.23003	0.261819	0.394957	-0.01197	0.012989	-0.01111	0.018495	0.046083	0.014595	tidak jatuh
8.562863	3.63368	-1.43963	0.164154	0.299007	1.0791	0.010163	-0.037	-0.01424	0.065362	0.155606	0.032879	tidak jatuh
-1.81457	8.989509	2.229241	0.882759	0.723145	1.281685	-0.10158	0.360563	0.100044	0.403879	0.467056	0.199587	tidak jatuh
-0.22841	4.890157	5.296692	3.419091	4.673911	3.522299	-0.10743	0.238337	0.197584	0.46332	0.737646	0.329469	tidak jatuh
-1.57898	2.388456	7.622181	2.677271	4.077669	3.266103	-0.04038	0.140774	0.073059	0.323531	0.589919	0.252883	tidak jatuh
-2.17082	0.902614	8.728302	2.139886	3.172727	2.449358	-0.01965	0.079328	0.019433	0.285234	0.43328	0.259374	tidak jatuh
2.43227	6.319495	2.213439	4.57704	3.705922	2.859251	-0.05575	0.012339	0.176021	0.402365	0.700059	0.475839	tidak jatuh
5.847119	4.654328	1.36661	4.742083	3.116176	2.189844	-0.05168	-8.32E-04	0.093913	0.326319	0.500084	0.409345	tidak jatuh
-0.89974	9.216719	1.230859	0.390176	0.2298	0.263872	-0.02644	-0.0716	0.01728	0.475745	0.309816	0.233619	tidak jatuh
-0.65529	9.460449	0.897347	0.381065	0.415634	0.420677	-0.01226	-0.02647	-0.27473	0.412048	0.654208	0.519328	tidak jatuh
-0.26815	9.497319	0.566708	0.398472	0.311082	0.430576	-0.00736	-0.03829	-0.12831	0.282106	0.472906	0.344438	tidak jatuh
-3.48308	2.637213	5.687426	1.577524	5.073867	4.101892	-0.22226	-0.04678	-0.05311	0.973646	0.29554	0.423811	jatuh
-2.31519	6.666415	1.987666	1.431726	5.464398	2.531285	-0.20015	-0.03445	-0.05508	0.80637	0.240724	0.519265	jatuh
-2.85724	4.931816	3.816118	1.976483	5.822698	3.983432	-0.02331	-0.01433	0.017387	0.604555	0.413423	0.384363	jatuh
-3.44358	2.195962	5.826529	2.183178	5.191686	4.230304	-0.1136	-0.09481	-0.10327	0.905236	0.321177	0.770217	jatuh
-3.51804	3.642538	4.947857	2.993974	5.976818	4.705852	-0.29828	-0.07235	-0.08447	1.156956	0.328455	0.512751	jatuh
-3.12108	5.035246	4.865497	1.727462	4.700156	3.832216	-0.14778	-0.0852	-0.06785	0.535027	1.233346	0.460918	tidak jatuh

-2.47369	6.531143	4.561433	1.553468	3.246097	3.727745	-0.06908	-0.3658	-0.13254	0.455936	0.57892	0.221712	tidak jatuh
-2.97144	4.784094	5.856936	1.264934	4.159827	3.834596	-0.10512	-0.18912	-0.08038	0.61253	0.445898	0.270094	tidak jatuh
-4.72496	-0.56312	8.904515	0.168627	0.371081	0.110375	-0.00553	-0.00453	0.015127	0.04378	0.017386	0.115311	tidak jatuh
-4.6821	-0.60693	8.919838	0.147661	0.231766	0.097056	-0.00151	0.005063	-0.0156	0.044614	0.054358	0.09983	tidak jatuh

LAMPIRAN B



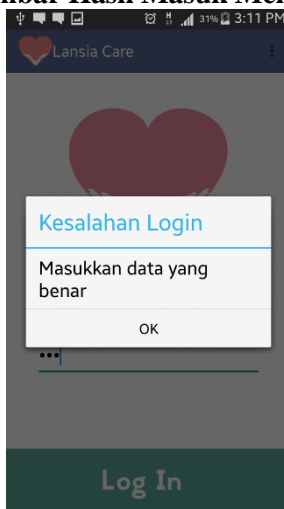
Gambar B. 1 Gambar Hasil Menu Awal Aplikasi



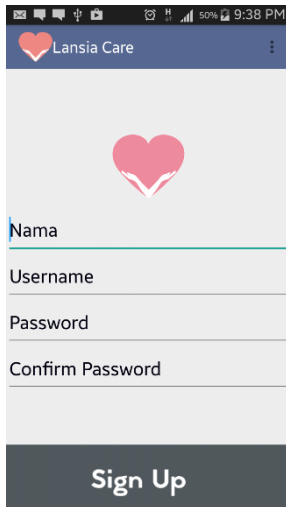
Gambar B. 2 Gambar Hasil Masuk Menu *Log In*



Gambar B. 3 Gambar Hasil Masuk Menu Utama Aplikasi



Gambar B. 4 Gambar Hasil Alert Dialog Kesalahan Login



Lansia Care

Nama

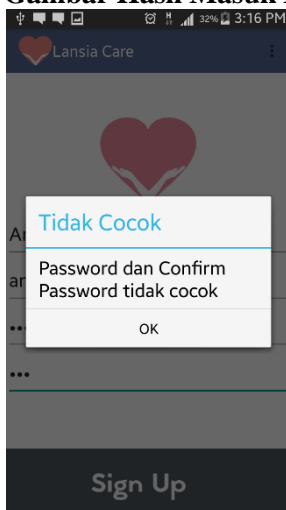
Username

Password

Confirm Password

Sign Up

Gambar B. 5 Gambar Hasil Masuk Menu *Sign Up*



Lansia Care

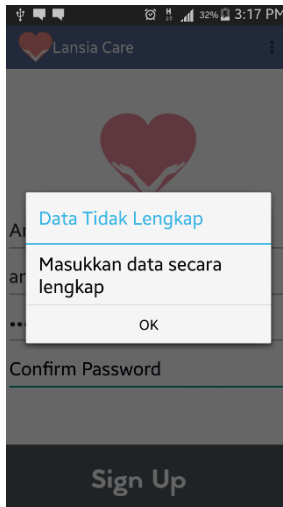
Tidak Cocok

Password dan Confirm Password tidak cocok

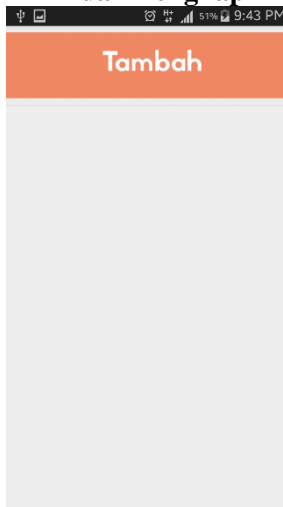
OK

Sign Up

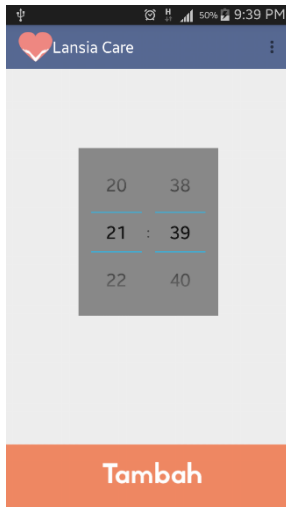
Gambar B. 6 Gambar Hasil *Alert Dialog* Kesalahan Ketidakcocokan



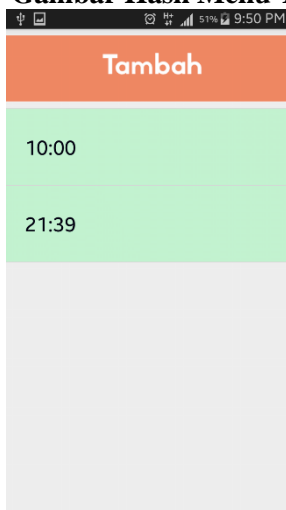
Gambar B. 7 Gambar Hasil Alert Dialog Kesalahan Data Tidak Lengkap



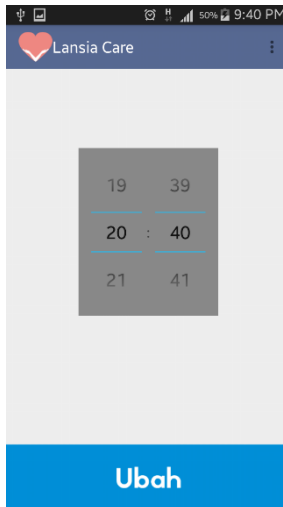
Gambar B. 8 Gambar Hasil Masuk Menu Atur Waktu Minum Obat



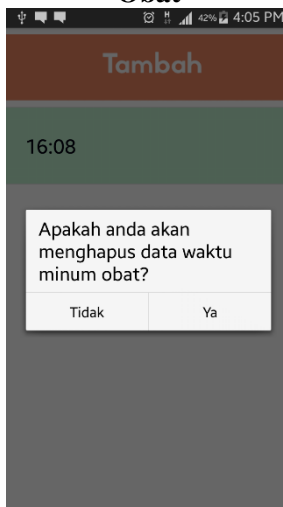
Gambar B. 9 Gambar Hasil Menu Tambah Obat



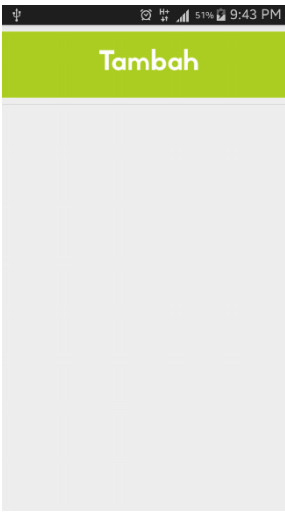
Gambar B. 10 Gambar Hasil Menampilkan Daftar Waktu Minum Obat



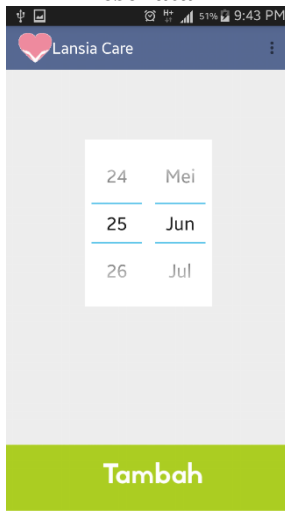
Gambar B. 11 Gambar Hasil Menu Ubah Waktu Minum Obat



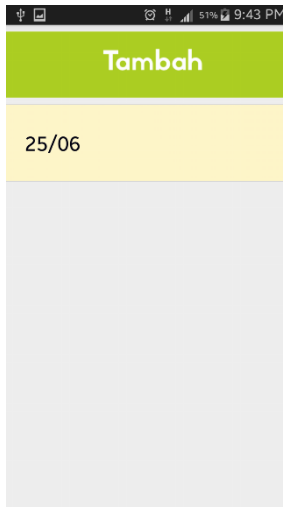
Gambar B. 12 Gambar Hasil *Alert Dialog* Hapus Waktu Minum Obat



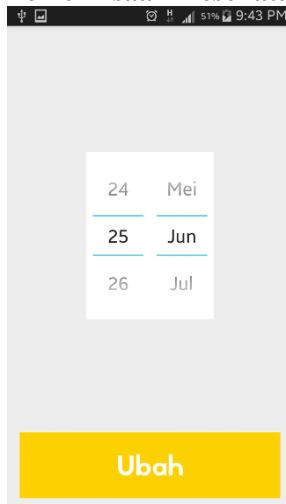
Gambar B. 13 Gambar Hasil Menu Atur Jadwal Pemeriksaan Kesehatan



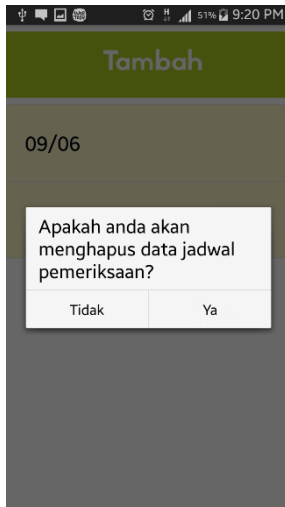
Gambar B. 14 Gambar Hasil Menu Tambah Waktu Pemeriksaan Kesehatan



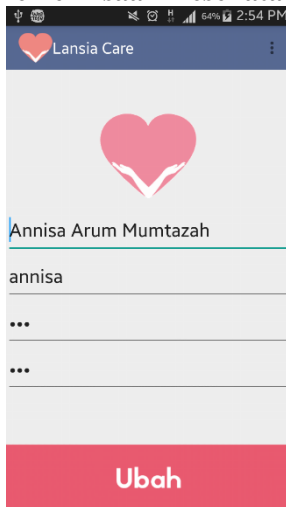
Gambar B. 15 Gambar Hasil Menampilkan Daftar Waktu Pemeriksaan Kesehatan



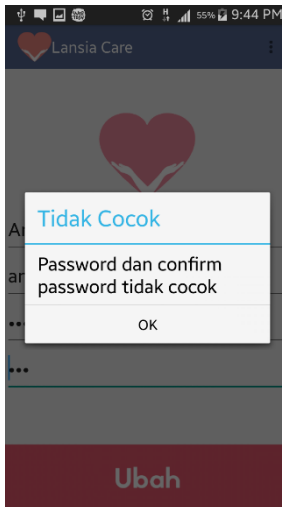
Gambar B. 16 Gambar Hasil Menu Ubah Waktu Pemeriksaan Kesehatan



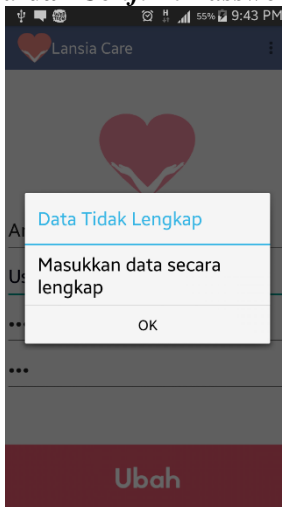
Gambar B. 17 Gambar Hasil Alert Dialog Hapus Waktu Pemeriksaan Kesehatan



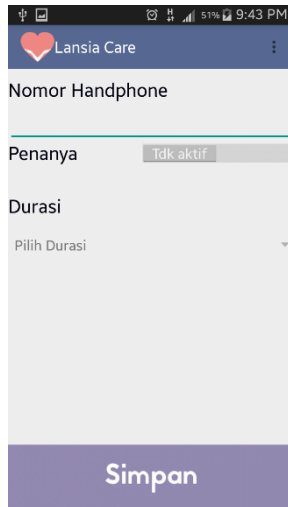
Gambar B. 18 Gambar Hasil Menu Ubah Profil Pengasuh



Gambar B. 19 Gambar Hasil *Alert Dialog* Kesalahan *Password dan Confirm Password Tidak Cocok*



Gambar B. 20 Gambar Hasil *Alert Dialog* Kesalahan Data *Pengasuh Tidak Lengkap*



Nomor Handphone

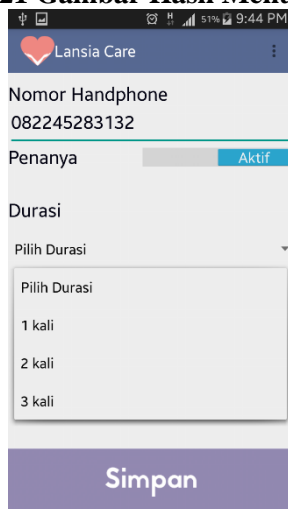
Penanya Tdk aktif

Durasi

Pilih Durasi

Simpan

Gambar B. 21 Gambar Hasil Menu Pengaturan



Nomor Handphone

082245283132

Penanya Aktif

Durasi

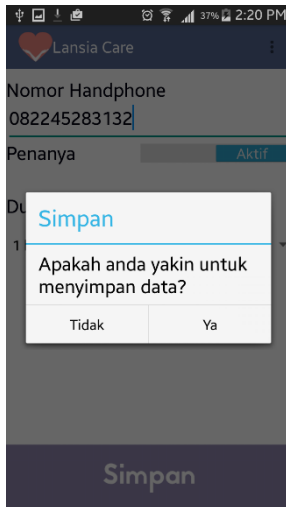
Pilih Durasi

Pilih Durasi

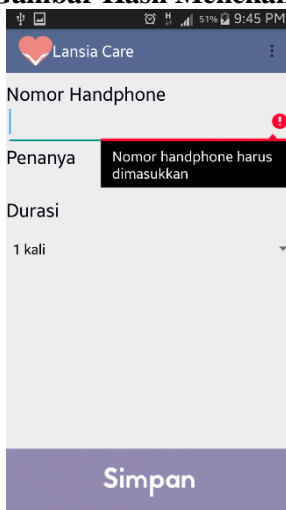
- 1 kali
- 2 kali
- 3 kali

Simpan

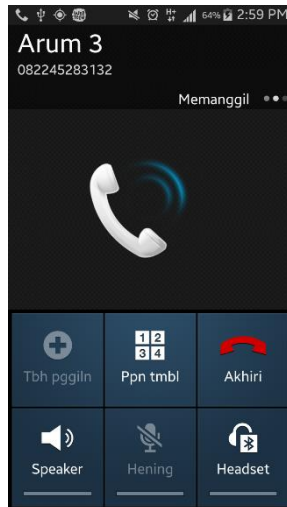
Gambar B. 22 Gambar Hasil Mengatur Durasi dan Nomor Handphone



Gambar B. 23 Gambar Hasil Menekan Pilihan Simpan



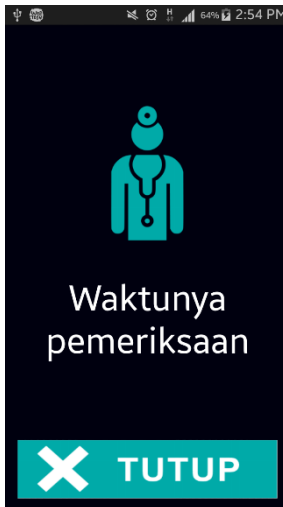
Gambar B. 24 Gambar Hasil *Error Text* Keharusan Pengisian Nomor *Handphone*



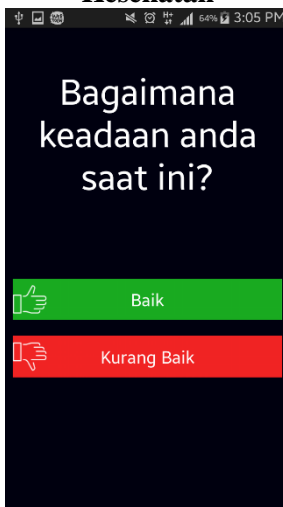
Gambar B. 25 Gambar Hasil Melakukan Panggilan Darurat Kepada Pengasuh



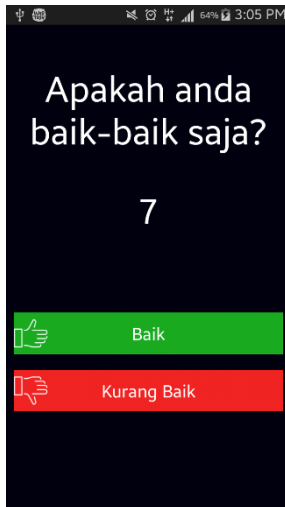
Gambar B. 26 Gambar Hasil Notifikasi Waktu Minum Obat



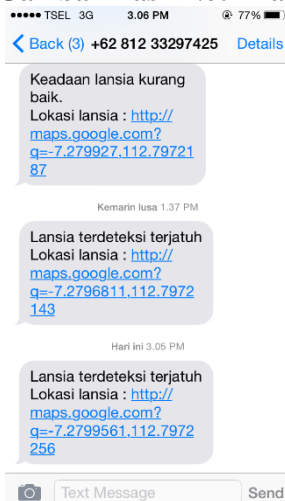
Gambar B. 27 Gambar Hasil Notifikasi Waktu Pemeriksaan Kesehatan



Gambar B. 28 Gambar Hasil Notifikasi Konfirmasi Keadaan



Gambar B. 29 Gambar Hasil Notifikasi Deteksi Jatuh



Gambar B. 30 Gambar Hasil Pesan Singkat pada *Handphone* Pengasuh

BIODATA PENULIS



Annisa Arum Mumtazah, biasa dipanggil Arum, lahir di Malang pada tanggal 28 Februari 1993. Penulis adalah anak kedua dari dua bersaudara. Penulis menempuh pendidikan mulai dari SDN Kalirejo5 Malang (1999-2005), SMPN 1 Singosari (2005-2008), SMAN 1 Malang (2008-2011) dan S1 Teknik Informatika ITS (2011-2015). Selama masa kuliah, penulis aktif dalam organisasi yang ada di lingkungan kampus ITS yaitu Himpunan Mahasiswa Teknik Computer-Informatika (HMTTC) dan Schematics ITS 2012. Penulis dapat dihubungi melalui *email*: annisamumtazah@gmail.com.